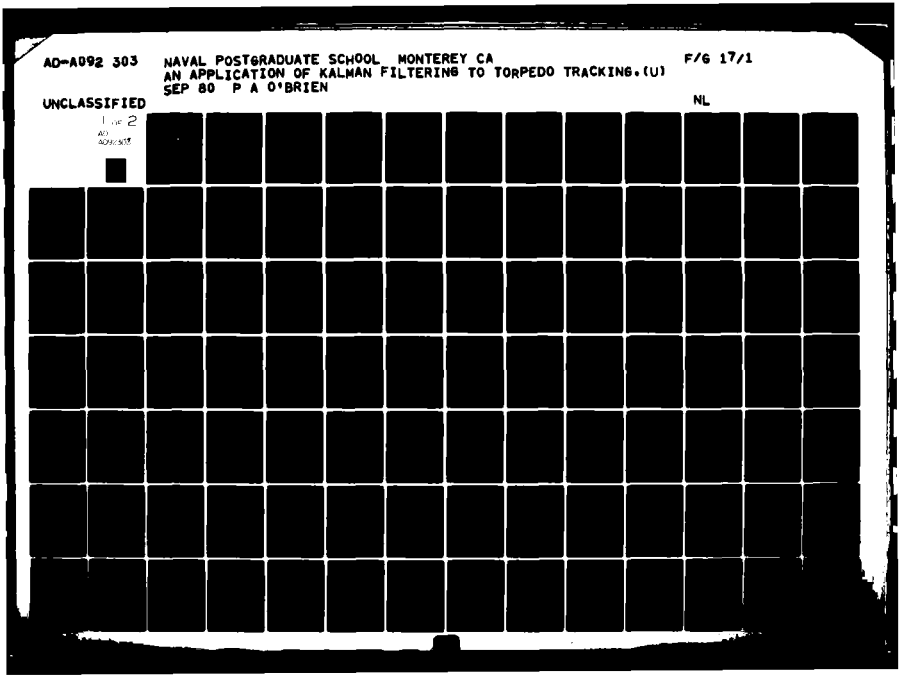
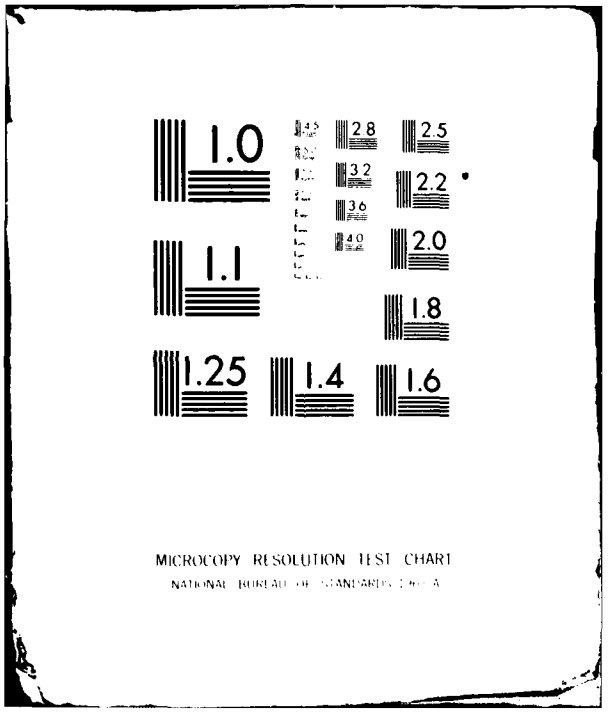
AD-A092 303 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 17/1
AN APPLICATION OF KALMAN FILTERING TO TORPEDO TRACKING.(U)
SEP 80 P A O'BRIEN

UNCLASSIFIED NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

AN APPLICATION OF KALMAN FILTERING
TO TORPEDO TRACKING

by

Paul A. O'Brien

September 1980

Thesis Advisor: H. A. Titus

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A092303	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN APPLICATION OF KALMAN FILTERING TO TORPEDO TRACKING.		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Paul A. O'Brien		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1980
		13. NUMBER OF PAGES 127
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Kalman Filtering
Torpedo Tracking

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A sequential Extended Kalman filter routine was developed to provide real time estimates of torpedo position and depth on the three dimensional underwater tracking range at the Naval Torpedo Station, Keyport, Washington. Filter measurements consisted of acoustic pulse transit times from the torpedo to a receiving hydrophone array. These measurements, which are nonlinear functions of the position and depth coordinates, were linearized and

DD FORM 1 JAN 73 1473 (Page 1) EDITION OF 1 NOV 68 IS OBSOLETE S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (continued)

and the filter gains calculated on-line. Tests were conducted using simulated torpedo trajectories that traversed both single and multiple hydrophone arrays. It was found that filter performance was dependent on system noise and the distance the torpedo was from the hydrophone array. Position and depth errors ranged between 0 and 10 feet.

Approved for public release; distribution unlimited

An Application of Kalman Filtering To Torpedo Tracking

by

Paul A. O'Brien
B.S., University of Maryland, 1972

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1980

Author Paul A. O'Brien

Approved by: H. A. Titus
Thesis Advisor

Alex Gerba Jr.
Second Reader

Chairman, Department of Electrical Engineering

William M. Tolles
Dean of Science and Engineering

ABSTRACT

A sequential Extended Kalman filter routine was developed to provide real time estimates of torpedo position and depth on the three dimensional underwater tracking range at the Naval Torpedo Station, Keyport, Washington. Filter measurements consisted of acoustic pulse transit times from the torpedo to a receiving hydrophone array. These measurements, which are nonlinear functions of the position and depth coordinates, were linearized and the filter gains calculated on-line. Tests were conducted using simulated torpedo trajectories that traversed both single and multiple hydrophone arrays. It was found that filter performance was dependent on system noise and the distance the torpedo was from the hydrophone array. Position and depth errors ranged between 0 and 10 feet.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author is deeply indebted to Professor Hal Titus for his counsel and professional guidance during this project.

I. INTRODUCTION

The Naval Torpedo Station (NAVTORPSTA) Keyport, Washington currently operates two three-dimensional (3-D) underwater tracking ranges with acoustical capability. The underwater tracking range utilizes a sonar transmitter installed in the torpedo to be tracked. The transmitter is synchronized with a master clock. Timed acoustic pulses are received by bottom mounted hydrophone arrays and then relayed via cable to a computer at the observation site. The computer calculates the positional coordinates of the torpedo and plots its trajectory through the water.

The measured data, which consists of the elapsed time from transmission of a pulse until its receipt at the hydrophone array, is corrupted with noise due to the combined effects of environmental factors and measurement instruments.

In order to achieve a more accurate plot of the torpedo path, editing and smoothing techniques have been applied to the data during post run analyses.

To improve the real-time capability of the 3-D tracking ranges, Kalman filtering theory has been applied to the computer generated X, Y, and Z positional coordinates but is not operational [1].

In the near future an updated computer system composed of three MODCOMPIV computers will be in operation. The

majority of the software development requires interfacing current tracking programs and other related programs to the new computer system.

An opportunity exists for expanding the capability of the tracking system by applying a Kalman filter routine which processes measurements of the transit times of the acoustic pulses and provides an estimate of the position of the tracked object in real time. Prior research in this area [2], revealed that a Kalman filter utilizing a sequential estimation approach was desirable.

The intention is to develop and test a sequential Kalman filter tracking algorithm that can be interfaced with the current underwater range system.

II. DESCRIPTION OF RANGE TRACKING GEOMETRY

The hydrophone array, consisting of four independent elements, defines an orthogonal coordinate system in which transit time measurements are made. As shown in Figure 1, four hydrophones X, Y, Z, and C are on four adjacent vertices separated by a distance d, along the edge of the cube. The origin of the array coordinates is at the center of the cube with the orthogonal coordinates parallel to its edge. Positional information is computed from the transit times of a periodic synchronous acoustic signal travelling from the torpedo to the four hydrophones on the array. The range measures the tracked torpedo's position every 1.31 seconds to an accuracy that is typically within 3 to 30 feet. A more detailed description of the range tracking capability is described in [3].

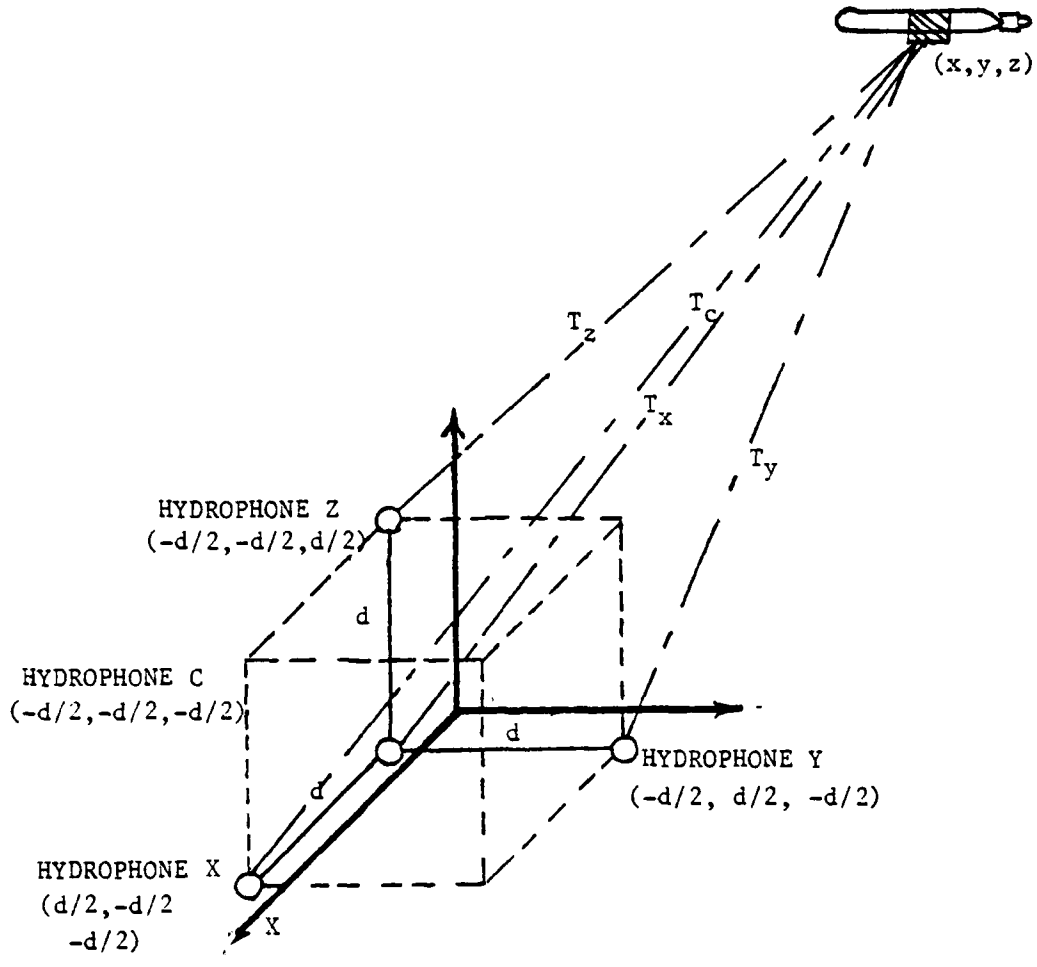


Figure 1. Geometry of a Tracking Array

III. THEORY--EXTENDED KALMAN FILTER

The equations describing the nonlinear relationship between transit times T_C, T_X, T_Y, T_Z and the position of the tracked torpedo are given by [3]

$$T_C = \frac{1}{VEL}\left[(x + \frac{d}{2})^2 + (y + \frac{d}{2})^2 + (z + \frac{d}{2})^2\right]^{\frac{1}{2}}, \tag{3.1}$$

$$T_X = \frac{1}{VEL}\left[(x - \frac{d}{2})^2 + (y + \frac{d}{2})^2 + (z + \frac{d}{2})^2\right]^{\frac{1}{2}}, \tag{3.2}$$

$$T_Y = \frac{1}{VEL}\left[(x + \frac{d}{2})^2 + (y - \frac{d}{2})^2 + (z + \frac{d}{2})^2\right]^{\frac{1}{2}}, \tag{3.3}$$

$$T_Z = \frac{1}{VEL}\left[(x + \frac{d}{2})^2 + (y + \frac{d}{2})^2 + (z - \frac{d}{2})^2\right]^{\frac{1}{2}}, \tag{3.4}$$

where d = 30 feet and VEL ≈ 4860, which is approximately the average velocity of propagation of sound in water at Dabob Bay.* Since the transit times were readily available and are nonlinear functions of position, these equations can be linearized and Kalman Filter theory applied using the Extended Kalman Filter [2]. This procedure producses a real time measurement, with filtering on the corrupted transit times T_C, T_X, T_Y, and T_Z without the necessity of converting these times to positions.

---

*Sound velocity in water varies as a function of temperature, salinity, and depth.

For the three-dimensional location problem three position states (x, y, z) and two velocity states (v_x, v_y) specify target motion. The discrete linear and nonlinear observation equations are given by

$$\underline{x}(k+1) = \Phi \cdot \underline{x}(k) + \Gamma \cdot \underline{w}(k) \tag{3.5}$$

and

$$z(k) = \underline{h}(x(k),k) + \underline{v}(k) \tag{3.6}$$

In these equations Φ and Γ are constant matrices and h is a nonlinear function of the state variable $\underline{x}$. $\underline{w}(k)$ is plant excitation noise and $\underline{v}(k)$ is measurement noise. The plant noise and measurement noise are assumed uncorrelated (white) with zero mean.

That is,

$$E[w(k) \cdot w^T(j)] = Q'(k)\delta_{kj}$$

and

$$E[v(k) \cdot v^T(j)] = R(k)\delta_{kj}$$

with

$$\delta_{kj} = 1, \quad k = j$$

$$= 0, \quad k \neq j$$

In order to apply the linear filter equation (3.6) is expanded in a Taylor series about the best estimate of the

the state at that time and only the first-order terms are kept.

Equation (3.6) gives

$$z(k) = H(k) \cdot \underline{x}(k) + \underline{v}(k) \tag{3.7}$$

where

$$H(k) = \left.\frac{\partial h}{\partial \underline{x}}\right|_{x'(k) = \hat{\underline{x}}(k/k-1)} \tag{3.7a}$$

$\hat{\underline{x}}(k/k-1)$ is the predicted value of the state before the k^{th} measurement.

A state error vector is defined by

$$\tilde{\underline{x}} = \hat{\underline{x}}(k) - \underline{x}(k),$$

and a predicted state error vector is defined by

$$\tilde{\underline{x}}(k/k-1) = \hat{\underline{x}}(k/k-1) - \underline{x}(k) .$$

The covariance of state error matrix is defined by

$$P(k) = E[\tilde{\underline{x}}(k) \cdot \tilde{\underline{x}}^T(k)],$$

and the predicted covariance of state error matrix is given by

$$P(k/k-1) = E[\tilde{\underline{x}}(k/k-1) \cdot \tilde{\underline{x}}^T(k/k-1)] .$$

The state excitation matrix is given by

$$Q(k) = \Gamma(k)E[\underline{w}(k) \cdot \underline{w}^T(k)] \cdot \Gamma^T(k)$$

and the measurement noise covariance matrix is

$$R(k) = E[\underline{v}(k) \cdot \underline{v}^T(k)] \quad .$$

The Kalman filter equations are given by [4,5]:

$$P(k+1/k) = \Phi P(k/k)\Phi^T + Q(k) \tag{3.8a}$$

$$G(k) = P(k/k-1)H^T(k)[H(k) \cdot P(k/k-1)H^T(k) + R(k)]^{-1} \tag{3.8b}$$

$$P(k) = [I - G/(k)H(k)]P(k/k-1) \tag{3.8c}$$

$$\hat{\underline{x}}(k+1/k = \Phi\underline{x}(k/k) \tag{3.8d}$$

$$z(k/k-1) = \underline{h}(\underline{x}(k/k-1),k)) \tag{3.8e}$$

$$\hat{\underline{x}}(k) = \hat{x}(k/k-1) + G(k)[z(k) - \underline{z}(k/k-1)] \tag{3.8f}$$

The Q matrix serves not only to allow for maneuvering but also to account for any model inaccuracies, that is, any discrepancies between the true action of the torpedo and its characterization by Equation (3.5). The Q also serves to prevent the gain matrix G(k) from approaching zero by always insuring uncertainty in the predicted covariance of error matrix P(k+1/k).

IV. PROBLEM DEFINITION--TORPEDO TRACKING WITH THE EXTENDED KALMAN FILTER

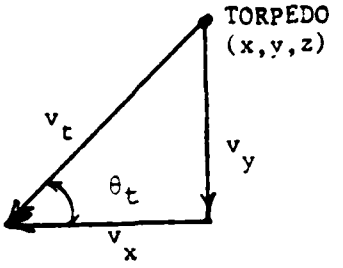
A. FILTER EQUATIONS

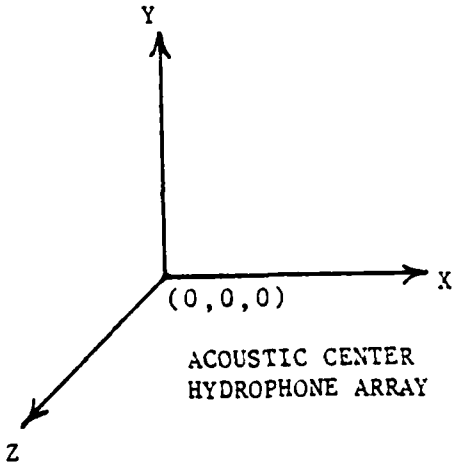
Figure 2 illustrates the geometry of the states used for the filter. For a constant course, constant-speed torpedo, the plant state equations can be described as two second order systems, one for X and one for Y. The depth (Z) of the torpedo is maintained constant and any velocity in the Z direction is assumed to be random, uncorrelated with zero mean. Z is therefore described as a first order system.

The plant state equations are

$$\begin{bmatrix} x(k+1) \\ v_x(k+1) \\ y(k+1) \\ v_y(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} x(k)+Tv_x(k)+g_1(\gamma_{\dot{\theta}_t},\gamma_{\dot{v}_t},k) \\ v_x(k)+g_2(\gamma_{\dot{\theta}_t},\gamma_{\dot{v}_t},k) \\ y(k)+Tv_y(k)+g_3(\gamma_{\dot{\theta}_t},\gamma_{\dot{v}_t},k) \\ v_y(k)+g_4(\gamma_{\dot{\theta}_t},\gamma_{\dot{v}_t},k) \\ z(k)+g_5(\gamma_{\dot{z}}) \end{bmatrix} \tag{4.1}$$

where $x(k)$, and $y(k)$ are the position coordinates at time $t(k)$, $v_x(k)$ and $v_y(k)$ are the X and Y components of velocity. T is the time between observations, i.e. $t(k+1)-t(k)$ and $\dot{z}(k)$ is the depth of the torpedo.





STATES

x

y

v_x

v_y

z

Figure 2. Filter Geometry

The excitation terms g_1 through g_5 are included to account for the random changes in speed, heading, and depth which can occur in a maneuvering torpedo. The quantities γ_{v_t}, $\gamma_{\dot{\theta}_t}$, and γ_z are the random changes of the torpedo which are assumed to be independent, zero mean, rates of change. They have variances defined by

$$\sigma_{\dot{v}_t}^2 = E[\gamma_{v_t}^2]$$

$$\sigma_{\dot{\theta}_t}^2 = E[\gamma_{\dot{\theta}_t}^2]$$

$$\sigma_{\dot{z}}^2 = E[\gamma_z^2]$$

The values of the standard deviations are taken from typical maneuvering parameters for the torpedo;

$$\sigma_{\dot{\theta}_t} = 22°/\text{sec}$$

$$\sigma_{\dot{v}_t} = 36 \text{ ft/sec}^2$$

$$\sigma_{\dot{z}} = 1 \text{ ft/sec}$$

The effect of this excitation is to increase the predicted value of the covariance of the state error matrix.

Writing the equations in state form resuts in

$$x(k+1) = \Phi x(k) + \Gamma w(k) \qquad (4.2)$$

where

$$\Phi = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\Gamma = \begin{bmatrix} T^2/2 & 0 & 0 \\ T & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}$$

The vector $\underline{w}(k)$ represents the effect on the states of the random excitations. Reference 5 has shown that the vector may be calculated from the equations relating torpedo X and Y velocity to the torpedo heading and velocity. The X velocity is

$$\dot{x} = v_x = v_t \cos\theta_t$$

which when differentiated gives

$$\ddot{x} = -v_t \dot{\theta}_t \sin\theta_t + \dot{v}_t \cos\theta_t \qquad (4.4)$$

where

$$\dot{\theta}_t = \gamma \dot{\theta}_t$$

$$\dot{v}_t = \gamma \dot{v}_t$$

$$\sin \theta_t = \frac{v_y}{v_t}$$

$$\cos \theta_t = \frac{v_x}{v_t}$$

and upon substitution of the above relationship in equation (4.4),

$$\ddot{x} = -v_y \gamma \dot{\theta}_t + \gamma \dot{v}_t \frac{v_x}{v_t} \tag{4.5}$$

Similarly

$$\dot{y} = v_y = v_t \sin\theta_t \tag{4.6}$$

$$\ddot{y} = v_t \dot{\theta}_t \cos\theta_t + \dot{v}_t \sin\theta_t$$

and after substitution

$$\ddot{y} = v_x \gamma \dot{\theta}_t + \frac{v_y}{v_t} \gamma \dot{v}_t \quad . \tag{4.8}$$

The depth term is just

$$z = \gamma \dot{z} \quad . \tag{4.9}$$

Thus

$$\underline{w}(k)^T = [(\gamma_{\dot{v}_t} \frac{v_x}{v_t} - \gamma_{\dot{\theta}_t} v_{\dot{y}}) \quad (\gamma_{\dot{\theta}_t} v_x + \gamma_{\dot{v}_t} \frac{v_y}{v_t}) \quad \gamma_{\dot{z}}] \tag{4.10}$$

where from the assumption on the γ's

$$E[\underline{w}(k)] = 0 \tag{4.11}$$

The excitation covariance matrix is thus found from

$$Q = \Gamma E[\underline{w}(k) w(k)^T] \Gamma^T \tag{4.12}$$

Let

$$\sigma_{\ddot{x}}^2 = (\frac{v_x}{v_t})^2 \sigma_{\dot{v}_t}^2 + v_y^2 \sigma_{\dot{\theta}_t}^2 \tag{4.13a}$$

$$\sigma_{\ddot{y}}^2 = (\frac{v_y}{v_t})^2 \sigma_{\dot{v}_t}^2 + v_x^2 \sigma_{\dot{\theta}_t}^2 \tag{4.13b}$$

$$\sigma_{\ddot{x}\ddot{y}} = v_x v_y [\frac{\sigma_{\dot{v}_t}^2}{v_t^2} - \sigma_{\dot{\theta}_t}^2] \tag{4.13c}$$

where the states are evaluated at the current state estimates $\underline{\hat{x}}(k)$. Substituting these expressions in the Q matrix results in

$$Q = \begin{bmatrix} (T^2/2)^2\sigma_{\ddot{x}}^2 & T^3/2\,\sigma_{\ddot{x}}^2 & (T^2/2)^2\sigma_{\ddot{x}\ddot{y}} & T^3/2\,\sigma_{\ddot{x}}\sigma_{\ddot{y}} & 0 \\ & T^2\sigma_{\ddot{x}}^2 & T^3/2\sigma_{\ddot{x}\ddot{y}} & T^2\sigma_{\ddot{x}\ddot{y}} & 0 \\ & & (T^2/2)^2\sigma_{\ddot{y}}^2 & T^3/2\sigma_{\ddot{y}}^2 & 0 \\ & & & T^2\sigma_{\ddot{y}}^2 & 0 \\ & \text{Symmetric} & & & T^2\sigma_{\dot{z}}^2 \end{bmatrix} \quad (4.14)$$

The excitation matrix serves not only to take into account the possibility of maneuvering, but of model inaccuracies as well [5]. Q is also used to prevent the gains of the filter from approaching zero as more and more data is processed, by insuring some uncertainty in the predicted state values.

The observation equations are nonlinear in the states and are given by

$$z(k) = \begin{bmatrix} T_C(k) \\ T_X(k) \\ T_Y(k) \\ T_Z(k) \end{bmatrix} = \begin{bmatrix} \frac{1}{VEL}[(x(k)+d/2)^2+(y(k)+d/2)^2+(z(k)+d/2)^2]^{\frac{1}{2}} + v(k) \\ \frac{1}{VEL}[(x(k)-d/2)^2+(y(k)+d/2)^2+(z(k)+d/2)^2]^{\frac{1}{2}} + v(k) \\ \frac{1}{VEL}[(x(k)+d/2)^2+(y(k)-d/2)^2+(z(k)+d/2)^2]^{\frac{1}{2}} + v(k) \\ \frac{1}{VEL}[(x(k)+d/2)^2+(y(k)+d/2)^2+(z(k)-d/2)^2]^{\frac{1}{2}} + v(k) \end{bmatrix} \quad (4.15)$$

The measurement noises, v(k)'s, are assumed to be zero-mean and independent with a covariance matrix

$$R(k) = \begin{bmatrix} \sigma_{T_C}^2 & 0 & 0 & 0 \\ 0 & \sigma_{T_X}^2 & 0 & 0 \\ 0 & 0 & \sigma_{T_Y}^2 & 0 \\ 0 & 0 & 0 & \sigma_{T_Z}^2 \end{bmatrix} \tag{4.16}$$

The magnitude of the transit times noise is a function of the signal-to-noise ratio at the hydrophone array, which is influenced by several environmental factors.

Discussions with NAVTORPSTA personnel indicate a typical accuracy of $\pm$ 10 microseconds and this value is used as the standard deviation for the simulation runs.

Equation (3.7a) can be used to give the linearized observation matrix. The result is

$$H(k) = \begin{bmatrix} \frac{\partial T_C}{\partial x} & \frac{\partial T_C(k)}{\partial v_x} & \frac{\partial T_C(k)}{\partial y} & \frac{\partial T_C(k)}{\partial v_y} & \frac{\partial T_C(k)}{\partial z} \\ \frac{\partial T_X(k)}{\partial x} & \frac{\partial T_X(k)}{\partial v_x} & \frac{\partial T_X(k)}{\partial y} & \frac{\partial T_X(k)}{\partial v_y} & \frac{\partial T_X(k)}{\partial z} \\ \frac{\partial T_Y(k)}{\partial x} & \frac{\partial T_Y(k)}{\partial v_x} & \frac{\partial T_Y(k)}{\partial y} & \frac{\partial T_Y(k)}{\partial v_y} & \frac{\partial T_Y(k)}{\partial z} \\ \frac{\partial T_Z(k)}{\partial x} & \frac{\partial T_Z(k)}{\partial v_x} & \frac{\partial T_Z(k)}{\partial y} & \frac{\partial T_Z(k)}{\partial v_y} & \frac{\partial T_Z(k)}{\partial z} \end{bmatrix} \tag{4.17}$$

When the derivatives are taken and evaluated at the predicted state values $x(k/k-1) = x'(k)$ the result is

$$H(k) = \begin{bmatrix} \frac{x'(k)+d/2}{DEN1} & 0 & \frac{y'(k)+d/2}{DEN1} & 0 & \frac{z'(k)+d/2}{DEN1} \\ \frac{x'(k)-d/2}{DEN2} & 0 & \frac{y'(k)+d/2}{DEN2} & 0 & \frac{z'(k)+d/2}{DEN2} \\ \frac{x'(k)+d/2}{DEN3} & 0 & \frac{y'(k)-d/2}{DEN3} & 0 & \frac{z'(k)+d/2}{DEN3} \\ \frac{x'(k)+d/2}{DEN4} & 0 & \frac{y'(k)+d/2}{DEN4} & 0 & \frac{z'(k)-d/2}{DEN4} \end{bmatrix} \quad (4.18)$$

where

$$DEN1 = [(x'(k)+d/2)^2+(y'(k)+d/2)^2+(z'(k)+d/2)^2]^{\frac{1}{2}}$$

$$DEN2 = [(x'(k)-d/2)^2+(y'(k)+d/2)^2+(z'(k)+d/2)^2]^{\frac{1}{2}}$$

$$DEN3 = [(x'(k)+d/2)^2+(y'(k)-d/2)^2+(z'(k)+d/2)^2]^{\frac{1}{2}}$$

$$DEN4 = [(x'(k)+d/2)^2+(y'(k)+d/2)^2+(z'(k)-d/2)^2]^{\frac{1}{2}} .$$

The Φ matrix, Q matrix, R matrix, and H matrix are then used in the Kalman filter equations (3.8).

B. THE SEQUENTIAL EXTENDED KALMAN FILTER

The Kalman filter equations described in the previous section represent a system that is characterized by a one-step estimation and prediction for each set of four transit time observables (T_C, T_X, T_Y, and T_Z). This one step process exhibits the following salient features:

1. The four transit times received during each 1.31 second measurement interval (time slot) are processed simultaneously by the filter equations. Updated covariance of error and state estimates are therefore obtained only once for each time slot.

2. A major part of the time period for the estimation calculation is spent in the gain equation where the inversion of a 4 x 4 matrix is required.

3. The Extended Kalman filter is based on a linearization about the predicted value of a period t_k to t_k+1. Unacceptably large or erroneous transit time measurements will cause the filter to ignore the update information for the entire time slot. The predicted estimate will become the new estimate for the time slot and the interval extended.

If one assumes that each of the four hydrophones in the array are statistically independent of one another, a sequential approach can be taken to process each of the four transit times separately. Thus, each observable will be processed in sequence, and the result of processing one measurement is used in the following computation to process the next measurement until all four observables have been used.

The advantages of the sequential approach over the traditional one-step estimation and prediction are:

1. Since the four transit times are independent and processed sequentially, the covariance of error matrix and

the state vector are updated four times during each time slot. Thus more accurate estimates of the filter states are achieved.

2. Modification of the filter equations for the sequential approach circumvented the matrix inversion in the gain equation. Therefore filter cycle time is reduced substantially.

3. An invalid transit time measurement will result in the filter ignoring the update information for that particular measurement only. For example, if only three of the four transit times are acceptable, the filter would be sequentially updated three times during that particular time slot. Thus, the filter is less sensitive to erroneous transit time measurements.

To utilize the sequential Extended Kalman filter, equations (3.8) must be modified. Calculations are performed on each of the four independent transit times in the following order: T_C, T_X, T_Y, and T_Z for each 1.31 second time slot. The estimate of the states x(k/k), based on one transit time measurement are used as the prediction x(k/k-1) for the calculations on the next measurement. Thus for the first time measurement T_C only the first row of the linearizing H matrix is calculated.

Next the first gain column corresponding to the first time measurement T_C is calculated by

$$G_{icol} = \frac{P(k/k-1)H_{irow}^T}{H_{irow}P(k/k-1)H_{irow}^T + R_{ii}} \qquad (4.19)$$

where i = 1 to 4 corresponding to the four measured transit times. Thus, the first row of the H matrix is used to calculate the first column of the gain matrix with both corresponding to the first measured time T_C.

Next, an estimate of the particular observation time is calculated using equation (3.8e) evaluated at the predicted state $\underline{x}(k/k-1)$.

The difference between observed transit time and the estimated transit times forms the residual which is used in the estimate equation

$$x_i = x(k/k-1) + G_{icol}[\text{Residual}] \tag{4.20}$$

This equation gives an estimate of the states based on one measurement.

Next, the covariance of error is calculated based on one measurement by

$$P_i = [I - G_{icol}H_{irow}]P_{i-1} \tag{4.21}$$

where

I = identity matrix

P_{i-1} = the covariance matrix calculated from the previous transit time measurement or if i = 1, the prediction P(k/k-1).

After the first iteration, $\bar{x}_1$ becomes $\underline{x}(k/k-1)$ and P_1 becomes P(k/k-1) for the second iteration which calculates the estimate of the states based on the second measurement T_X.

After four iterations (k = 4), $\underline{x}_4$ becomes the estimate for the time slot x(k/k) and P_4 becomes the covariance of error P(k/k).

The predictions for the next time slot are calculated using equations (3.8a) and (3.8d). This process is repeated for each time slot.

IV. TESTING AND SIMULATION

A. DESCRIPTION

The sequential Extended Kalman filter routine is tested using simulated torpedo tracks generated by the IBM-360 computer. A variety of track scenarios are produced to test the filter performance during single array tracking.

The first series of tests used constant course and speed tracks which transitioned multiple array quadrants including:

1. crossing north of the array
2. crossing above the array
3. crossing diagonally through the array

The next series of tests adds a number of typical torpedo maneuvers to the constant course and speed tracks. These maneuvers consisted of 10 and 20 °/sec turns with G forces ranging from 1/4-G to 1-G.

For the above tests filter initialization errors range from 0 to 40 feet/sec for X and Y velocity, and 0 to 60 feet for X and Y position. The Z depth is maintained constant at 300 feet which is approximately the average depth of water at Dabob bay. Zero mean Gaussian noise is added to corrupt the observed transit times for all runs.

The last series of tests demonstrates the ability of the filter to track through the areas of multiple arrays.

Before the above series of tests can be conducted, it is necessary to add operational features to the software which

provide the following capability:

1. detect erroneous transit time measurements
2. minimize position error during high dynamic maneuvers
3. track through multiple arrays

A more detailed explanation of these features is described below.

B. EDITING ERRONEOUS TIME MEASUREMENTS

The operation of the filter may be adversely affected by erroneous transit time measurements. Large errors in the transit time measurements will produce errors in the state vector estimate and the linearizing H matrix. This may cause the filter to become unstable and exhibit divergent characteristics. In some extreme cases catastrophic failure may result. Measurement errors can occur because of many factors including an error in the transit time of the acoustic pulse primarily due to the receipt of multi path signals from previous time slots that have reflected off the surface, bottom, or different density layers in the water.

To guard against catastrophic failures a three-sigma gate using the covariance of measurement noise (R) and the covariance of estimation error (P(k/k)) is implemented. For each calculation of a state estimate ($\underline{x}(k/k)$), the largest positional covariance of error is used, either $\underline{x}$, $\underline{y}$, or $\underline{z}$, and converted to time in seconds using the average velocity of sound in water for Dabob bay, 4860 ft/sec. The gate then is

written for each time measurement i = 1 to 4:

$$\text{Gate} = \frac{\sqrt{(P_{jj})_{\text{maximum}} + R_{ii}}}{4860}$$

where j = 1, 3, 5. The gate expands or decreases depending on the confidence level of the position estimate and the transit time. If ZDIFF which is the difference between the actual transit time received and the predicted transit time to a particular hydrophone exceeds the gate, the measurement is considered unacceptable and the filter gain is set to zero causing the filter to ignore the data and take the prediction of the states as the estimate

$$\underline{x}(k/k) = \underline{x}(k/k-1)$$

An invalid time measurement zeros only the gain column for that particular hydrophone causing only that hydrophone's data to be ignored.

C. BOUNDING RESIDUAL BIAS ERROR

As previously mentioned, the torpedo dynamics are modeled in the filter's excitation covariance matrix Q.

During preliminary tests it becomes evident that torpedo turn rates between 5 and 20°/sec yield position errors ranging from approximately 10 to 40 ft. Increasing the standard deviation parameters ($\sigma_{\dot{\theta}}$ σ_{v_t} $\sigma_{\dot{z}}$) for the adaptive Q matrix helps to a certain extent, however as expected a point is reached where the noise starts to become the dominant error source. There are two major reasons why the filter develops errors during turns:

1. The filter's predicted state estimates are computed assuming a constant velocity over a 1.31 second sampling interval. During a turn the velocity vector of the torpedo is continuously rotating, therefore the filter's predicted estimate will be in error. A large magnitude error will require more than one time slot to remove. A faster sampling rate would reduce the error and may alleviate the problem; however, the Dabob tracking range system has been designed for a 1.31 second sampling rate and cannot be changed.

2. The adaptive Q matrix assumes that the torpedo accelerations are normally distributed and uncorrelated. Torpedo tracking maneuvers do not fit these assumptions.

To minimize position errors during maneuvers a procedure has been developed in order that repeated estimates of the state vector are obtained during a single sampling period (1.31 second time slot) until the transit time residual error falls below a preset threshold.

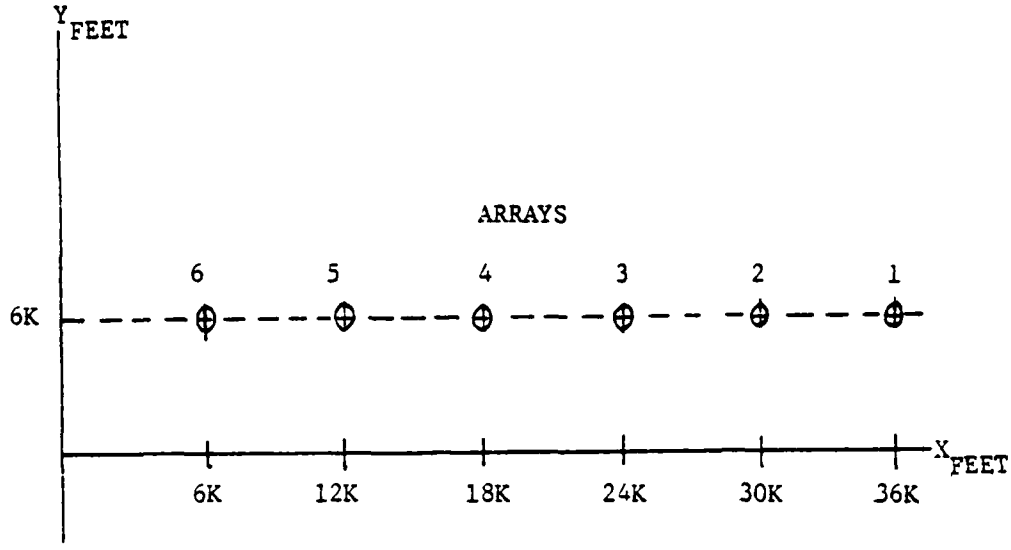
In Section IV it was stated that during each time slot the gain, covariance, and state estimate equations (4.19-4.21) were iteratively solved for each of the four sequential transit times T_C, T_X, T_Y, and T_Z. Subsequently additional software was incorporated to calculate the estimated transit time to each of the four hydrophones based on the last updated state vector. Four time residuals are then computed by subtracting the estimated time to each hydrophone from the observed transit time. The average of the absolute value of

the time residuals is then formed. If the average residual error exceeds the preset threshold in the software, which is indicative of large error in the state vector, Q is calculated and added to the last updated covariance of error matrix P. The filter then reiterates the gain, covariance, and state estimate equations for the same time slot. If the new state vector generates an average residual error still exceeding the threshold value, then the process is repeated. This procedure continues until the average residual error falls below the preset threshold at which time an acceptable state vector estimate has been obtained for the time slot. Once this occurs the prediction equations (3.8a) and (3.8d) are solved and the filter proceeds to the next time slot and commences processing new transit time measurements. As will be shown later the filter did successfully bound the position error during turning maneuvers. For these tests the threshold constant is set at 10 microseconds, which is the value of RMS measurement noise.

C. MULTIPLE ARRAY TRACKING

Initial tests were performed on tracks in the area of one array. In order to more closely simulate a typical run on the range, a scheme was designed to track the torpedo through multiple arrays [2].

First, a coordinate system is defined as shown in Figure 3. The center of the coordinate system is geographically



Coordinate System for Multiple Array Tracking

ARRAYS	C HYDRO			X HYDRO			V HYDRO			Z HYDRO		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
	36000	6000	0	36030	6000	0	36000	6030	0	36000	6000	30
	30000	6000	0	30030	6000	0	30000	6030	0	30000	6000	30
	24000	6000	0	24030	6000	0	24000	6030	0	24000	6000	30
	18000	6000	0	18030	6000	0	18000	6030	0	18000	6000	30
	12000	6000	0	12030	6000	0	12000	6030	0	12000	6000	30
	6000	6000	0	6030	6000	0	6000	6030	0	6000	6000	30

HYDRO--Hydrophone Location Matrix

Figure 3

near the entrance to Dabob bay in the simulation. Array number 6 is the closest array to the coordinate center. In the simulation array 1 is at 36,000 feet from coordinate center and array 6 is 6,000 feet. The C hydrophone is assumed to be the axis location of each array. Then each X position for the X hydrophone in each array is $X_C + 30$, each Y position for the Y hydrophone is $Y_C + 30$ and each Z position for the Z hydrophone is $Z_C + 30$. These 72 positions, an XYZ position for each of 4 hydrophones in 6 arrays, are placed into a 6 x 12 matrix HYDRO and referenced throughout the routine. The torpedo position is referenced to a central level rectangular coordinate system. The nonlinear observation equations become

$$z(k) = \begin{bmatrix} T_C(k) \\ T_X(k) \\ T_Y(k) \\ T_Z(k) \end{bmatrix} = \begin{bmatrix} \frac{1}{VEL}[(x'(k)-X_{iC})^2+(y'(k)-Y_{ic})^2+(z'(k)-Z_{iC})^2]^{\frac{1}{2}} \\ \frac{1}{VEL}[(x'(k)-X_{iX})^2+(y'(k)-Y_{iX})^2+(z'(k)-Z_{iX})^2]^{\frac{1}{2}} \\ \frac{1}{VEL}[(x'(k)-X_{iY})^2+(y'(k)-Y_{iY})^2+(z'(k)-Z_{iY})^2]^{\frac{1}{2}} \\ \frac{1}{VEL}[(x'(k)-X_{iZ})^2+(y'(k)-Y_{iZ})^2+(z'(k)-Z_{iZ})^2]^{\frac{1}{2}} \end{bmatrix}$$

where $T_C(k)$, $T_X(k)$, $T_Y(k)$, and $T_Z(k)$ are the transit times to each of the four hydrophones and X'(k), Y'(k), and Z'(k) are the filter estimates x'(k/k) evaluated at the predicted positions $\underline{x}(k/k-1)$. The subscripted variables X, Y, and Z are the coordinates of a particular array being used.

The decision parameter used to determine the switching from array to array is a straight handoff. If the predicted x position is greater than 3,000 feet from the array in use, then an index (I8) is incremented and the next row of HYDRO is implemented. This placed into the routine the X, Y, and Z positions of the hydrophones in the next array. The handoff can easily be utilized in real range operations, as the transit times from adjacent arrays are present at the computer for a particular time slot. For simulation, it is assumed that in all the arrays each axis pointed in the same direction. In actual range operations each array is tilted about both the X and Y axis. Since the true transit times are derived in a tilted coordinate frame, the filter's estimate of transit time must also be calculated in a tilted coordinate frame. The tilt angle measurements along with the level rectangular coordinates of the array with respect to the central rectangular coordinate system can be input into the matrix HYDRO to rotate the coordinates of each hydrophone in the array.

VI. TEST RESULTS AND DISCUSSION

A. SERIES ONE--STRAIGHT RUNS

This series of tests includes constant speed and heading tracks. Initial position and velocity errors are set to zero. Measurement noise is added to all runs. Torpedo velocity ranged from 20 to 55 knots. Figure 4 shows the true trajectory of the torpedo in the horizontal X-Y plane. The center of the hydrophone array is located at positional coordinates (0,0). Figure 5 describes the filter's RMS estimate of position error as a function of sampling time. The RMS estimates are obtained by taking the square-root of the appropriate diagonal terms of the covariance matrix. It should be noted that the propagation of RMS estimates are a function of the position of the torpedo with respect to the hydrophone array. As shows in Figure 4, if Y = 0, the torpedo path is inbound toward the array along the "X" axis. Upon examining Figure 5, it is evident that the RMS estimate of X position error or the uncertainty in the filter's X position estimate is small and relatively constant throughout the entire trajectory. The uncertainty in the filter's Y position estimate decreases as the torpedo approaches the array, becomes a minimum at the closest point of approach (CPA), and increases as the torpedo moves outbound away from the array. This behavior in the filter's RMS estimates can be explained by recognizing

that the cluster of four hydrophones in the array actually describe a position fixing system. The torpedo position is located by the intersection of circular lines of position (range arcs) originating from each hydrophone on the array. In general, the greater distance the torpedo is located from the array, the shallower the range arc intersections become and the more uncertainty exists in the actual torpedo position. For the case where the torpedo is moving along the X axis, the uncertainty in position exists only in the Y direction. If the torpedo's X distance from the array becomes infinite, the range arcs from each hydrophone appear to originate from a single point source. The position fixing system then reduces to a single dimension ranging system along the X axis with the uncertainty in the Y position estimate becoming infinite. If the torpedo approaches the array along the Y axis (X = 0), the filter's RMS estimate of Y position error will be relatively small, and as the torpedo's Y distance from the array becomes infinite, the filter's uncertainty in the X position estimate becomes infinite.

In order to keep RMS position error estimates within allowable tolerance the Dabob range tracking system requires the torpedo remain within a 3,000 foot radius of the center of the hydrophone array.

Figures 6 and 7 depict the error in both position and depth for the straight run trajectory along the X axis. The position errors are computed by subtracting the filter

position estimate, x(k/k), from the computer generated true position for each time slot. From Figure 6, it is observed that the Y position error magnitude, which is caused by system noise, decreases on the inbound leg and increases on the outbound leg. Since the torpedo path originated on the outer tracking perimeter (X = 3,000 feet), the noise in the Y channel will be amplified due to the large uncertainty in the Y position estimate. The magnitude of noise error in the Y direction also decreases as the torpedo approaches the array and increases as the torpedo moves outbound. The steady state X and Y position errors ranged between -11 and +4 feet throughout the trajectory.

Figure 8 describes a straight inbound torpedo path that passes north of the array at a constant Y distance of 2121 feet. Torpedo velocity is maintained at 20 knots. A comparison of the RMS x estimates of this track with the preceding one, Figures 5 and 9, reveal that both estimates are relatively constant, however the X estimate in Figure 9 is significantly larger. Since the torpedo path is parallel to the X axis, a larger uncertainty in the X position estimate will result. A comparison of the RMS Y position error estimates reveals identical propagation characteristics. This is due to the symmetry in the east-west direction of the two trajectories.

Figures 10 and 11 show that the steady state position and depth magnitude errors are less than 10 feet. The noise

induced error in the X position is due to the uncertainty in the filter X position estimate.

A torpedo path which is inbound along a 45° diagonal to the array is depicted in Figure 12. For this run torpedo velocity is constant at 28 knots. Figure 13 shows that the filter's X and Y RMS estimates of position error are approximately equal. Likewise Figure 14 shows that X and Y position error propagation is also similar. These results are consistent since the X and Y position magnitudes of the torpedo trajectory are identical. At the point of closest approach (X = 0, Y = 0) the uncertainty in both the X and Y positions becomes a minimum and increases as the torpedo moves past the array. Steady state position and depth errors range between -4 and +6 feet.

B. SERIES TWO--MANEUVERING RUNS

The results of the straight run analyses discussed in the previous subsection show that the propagation of the filter's estimate of RMS position error is dependent on the path of the torpedo with respect to the hydrophone array. Thus the erros in X and Y position, and Z depth are also a function of the torpedo trajectory. In this series of tests maneuvers are added to the three straight path trajectories. These maneuvers consist of 360° turns at turn rates of 10 and 20°/sec. G forces range between 1/4-G and 1-G. G forces are changed using the same turn rate by varying velocity. A typical test

run on the tracking range at Dabob would include these type maneuvers.

Figure 16 describes an inbound torpedo path which maintains a Y distance from the array between -200 and +600 feet. Initial position and velocity are 30 and 20 feet in X and Y position and 30 and -20 feet/second in X and Y velocity. Torpedo velocity is maintained constant at 20 knots throughout the trajectory. Four 360° 1/4-G turns are executed during the run. Figures 17 and 18 describe the position and depth errors. Within approximately three time slots the filter achieved steady state performance. Upon observing the error propagation it is apparent that the filter is insensitive to turn rate maneuvers. Negligible degradation in position and depth is observed. Thus the error bound routine discussed in the previous section successfully bounds position and depth errors during turns. As expected the position errors decrease as the torpedo approaches the array and increase as the torpedo proceeds outbound from the array. Since position and depth errors exhibit approximately equal oscillations about zero indicates that the system noise is the dominant error source driving the filter. Bias errors on the time residuals due to maneuvers are negligible. Steady state errors in position and depth ranged between -1 and +5 feet throughout the run.

The next test describes an inbound torpedo trajectory, Figure 19, which passes north of the hydrophone array. The torpedo maintained a Y distance from the array between 1600

and 2400 feet. Four 10°/sec 1/4-G turns are executed during this test. Filter initialization errors are 20 and 11 feet in X and Y position and 20 feet/sec in both X and Y velocity. The error in position and depth is described in Figures 20 and 21. The X position error demonstrates larger deviations than the Y position error. As noted during the straight runs, this is due to the difference in the filter's X and Y RMS position error estimates for this type trajectory. The oscillatory characteristics of the position error indicate again that noise is the dominant error source and the filter is tracking the torpedo through the maneuvers. Position errors range between -8 and +8 feet with one X channel error reaching a maximum of -15 feet, which occurred at time slot 140. The exact cause of the -15 foot error is unknown but may have been caused by an anomaly in the IBM-360 subroutine SNORM which generates the zero mean Gaussian noise for the filter. Errors in torpedo depth, Figure 21, are consistent and range between -2 and +5 feet.

For the next test the torpedo exhibits an inbound trajectory along a 45° diagonal with respect to the array. Figure 22 shows that four 10°/sec 1/4-G turns are executed on this run: 2 on the inbound leg and 2 on the outbound leg. Filter initialization errors are 60 feet in both X and Y position and 30 feet in X and Y velocity. Steady state conditions are achieved in approximately four time slots. As expected for this type trajectory, the X and Y position

errors portray similar propagation characteristics, Figure 23. This similarity can be explained by the equivalent X andY RMS error estimates generated by the filter for a 45° diagonally inbound trajectory. Position and depth errors, Figure 24, rnged ranged between -7 and +10 feet.

The preceeding tests discussed filter performance for straight runs with 10°/sec 1/4-G turns. Using the same basic torpedo trajectories as above, similar tests are performed with 10°/sec turn rates at 1/2-G. Figures 25 through 33 describe the torpedo paths and the corresponding position and depth error propagation plots for each run. Filter initialization errors ranged between 10 and 40 feet in position and 10 and 40 feet in velocity. Torpedo velocity is maintained constant at 55 knots. An analysis of the position and depth plots reveals that filter performance is similar to that described in the previous tests. In Figures 26 and 32 the larger errors incurred after time slot 135 are due to the torpedo trajectory exceeding the radial tracking range limit of 3,000 feet. This is more clearly described in the torpedo trajectory curves, Figures 25 and 31. The additional 1/4-G acceleration during the turns did not produce visible degradation in the filter's position and depth estimates.

The next set of maneuvering tests increased the turn rate to 20°/sec with G forces at 1/2-G. The same basic inbound and outbound straight runs trajectories are used. Initialization errors ranged between 20 and 40 feet in position and -10 and

and 35 feet/second for velocity. Figure 34 describes a torpedo inbound trajectory which maintains a Y distance from the array between -275 and 500 feet. Four turns are executed on this run. The X and Y position errors reveal essentially the same characteristics described before. In general the position and depth errors decreased as the torpedo approaches the array, become a minimum at the closest point of approach, and increase as the torpedo moves outbound past the array. The higher turn rates experienced on this test do not degrade the X and Y position errors. Figures 35 and 36 show that position and depth errors ranged between -1 and +5 feet.

Figure 37 depicts a torpedo path which maintains a Y distance from the array between 1760 and 2760 feet. Three 20°/sec 1/2-G turns are performed during this run. Figures 38 and 39 describe the position and depth errors which range between -7 and +9 feet during steady state.

The diagonally inbound trajectory with four 20°/sec 1/2-G turns is displayed in Figure 40. Position and depeth errors for this run, Figures 41 and 42, ranged between +10 and -10 feet. Again note the similarity in the position error behavior for each channel for this type trajectory. The larger errors in position after time slot 125 are due to the filter's RMS position error estimates increasing as the torpedo approaches the radial tracking limit of the hydrophone array.

The last set of maneuvering tests increased the G force during the 20°/sec turns from 1/2-G to 1-G. The three basic

inbound-outbound trajectories are again repeated. Figures 43 through 51 describe the torpedo trajectories and corresponding position and depth errors. Overall performance indicates that during steady state conditions the magnitude of position and depth errors are below 10 feet.

Due to the presence of system noise position errors reached a maximum of 14 feet, when the torpedo trajectory exceeded the radial tracking limit of the hydrophone array. As was also observed in the preceding tests, errors induced by the turn rate appeared to have a negligible effect on filter performance.

C. SERIES THREE--MULTIPLE ARRAY TRACKING

The purpose of the last series of tests is to functionally demonstrate that the filter is able to track the torpedo through multiple arrays using the simulated tracking range set up and handoff scheme described in the previous section. Figure 52 depicts a straight run trajectory which traverses the six hydrophone arrays. The position and depth errors are described in Figures 53 and 54.

Note that the handoff from array to array was accomplished with negligible effect on position and depth errors. The magnitude of the errors compare well with those observed on the single array straight run tracking.

VII. CONCLUSIONS

The sequential Extended Kalman filter satisfactorily provided real time estimates of torpedo position and depth. The magnitude of steady state position and depth errors ranged between 0 and 10 feet for torpedo tracks within the specified radial tracking range.

The iterative error bound routine furnished the capability to estimate torpedo position and depth during turning maneuvers up to 1-G. Errors generated during maneuvers are comparable to those observed during straight runs.

The filter's RMS estimates of position and depth errors are dependent on system noise and the distance the torpedo is from the center of the hydrophone array. In general larger errors are encountered at more remote distances from the hydrophone array.

The simulated trajectories were performed using constant measurement noise sequences. Since the filtering problem is non-linear the error, in the filter's state estimate vector is a function of the statistical properties of the noise. Future tests should include a Monte Carlo simulation where filter performance is investigated for an ensemble of measurement noise sequences.

Additional tests should also include evaluating filter performance using trajectories generated from actual torpedo

runs on the Dabob test range. These tests would verify the adequacy of the noise model in the filter and the ability of the software to edit erroneous transit time measurements.

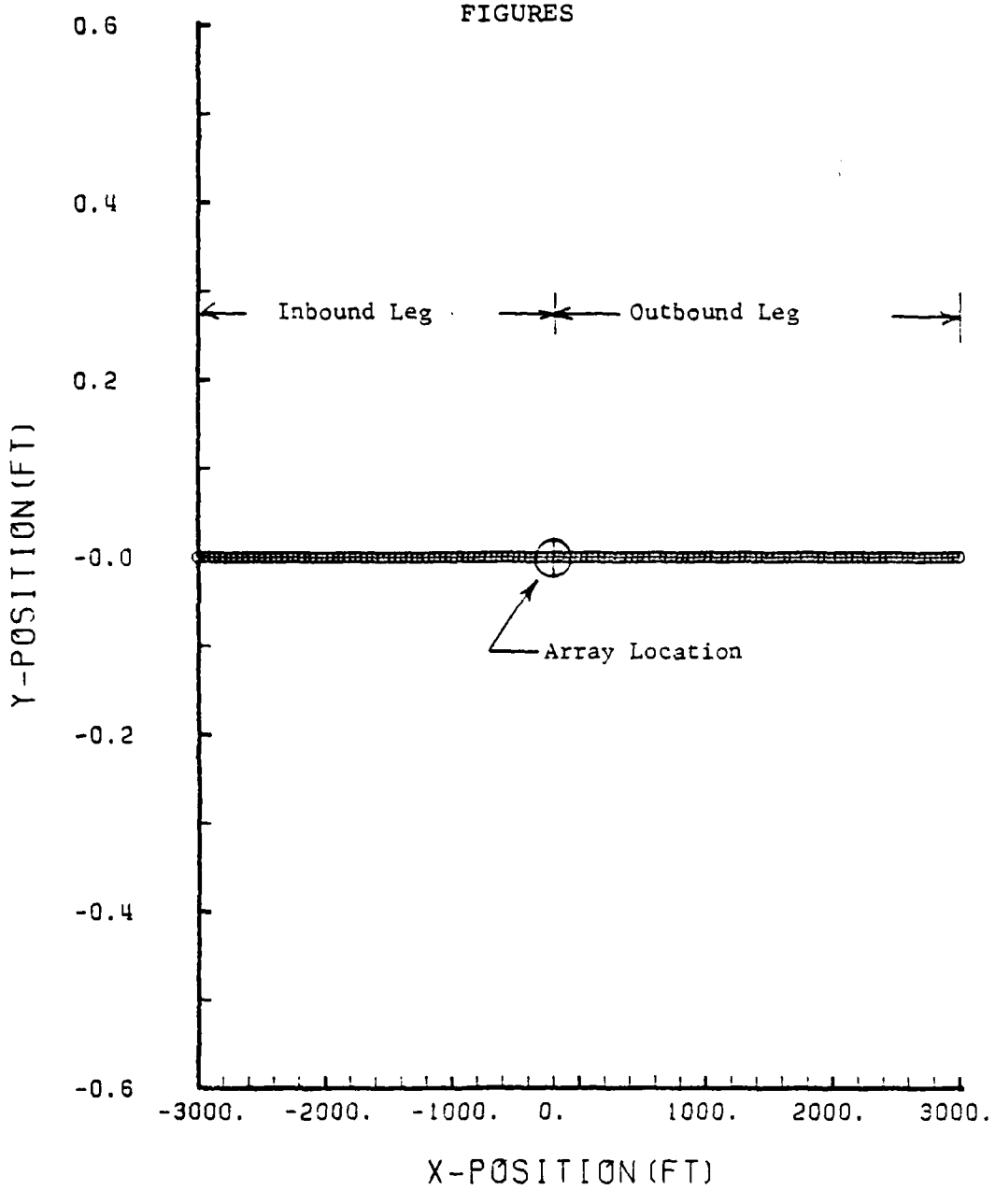


Figure 4. True Trajectory of the Torpedo in the Area of a Single Array

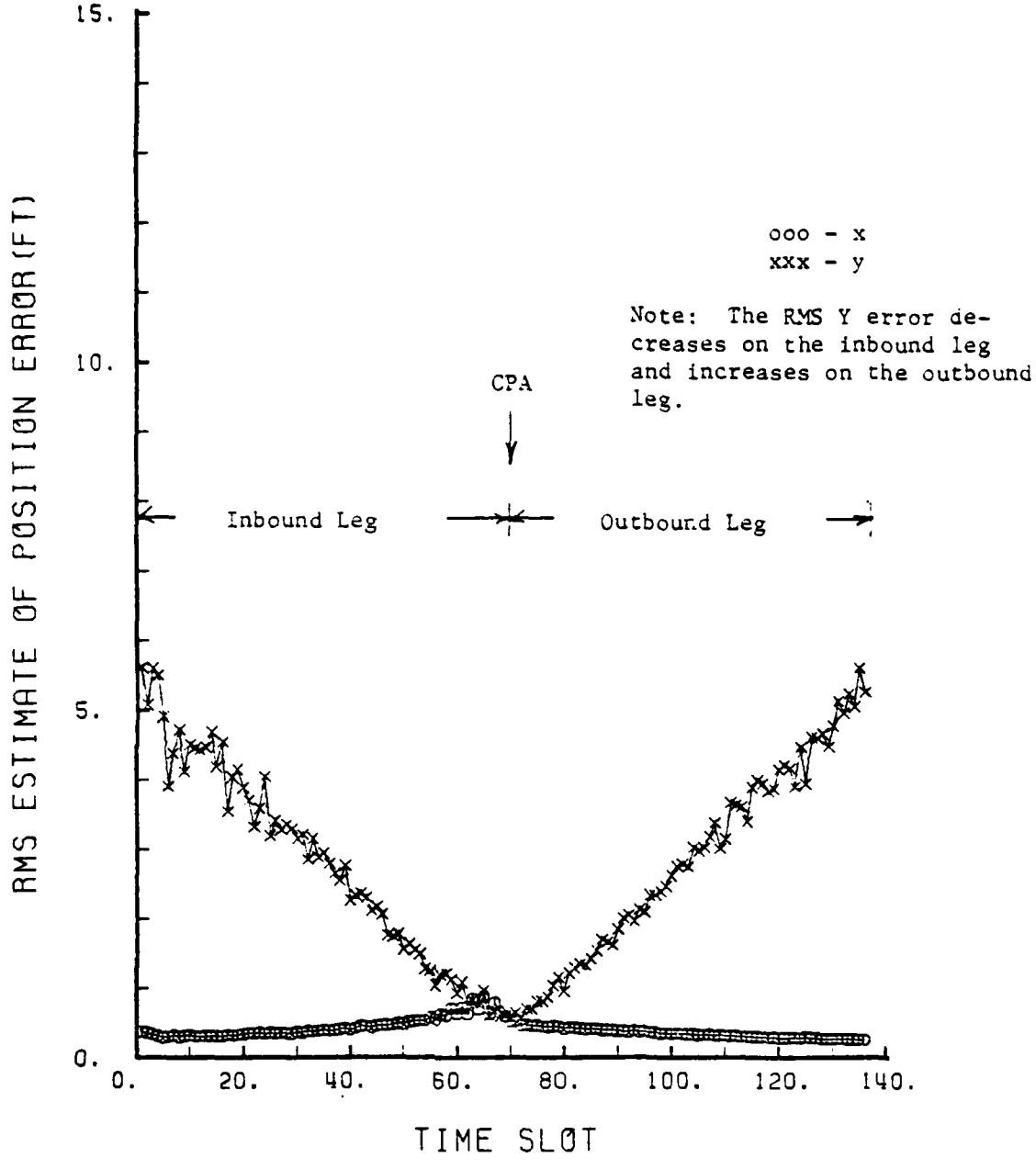


Figure 5. Filter Estimate of RMS Position Error During a Straight Run in the Area of a Single Array

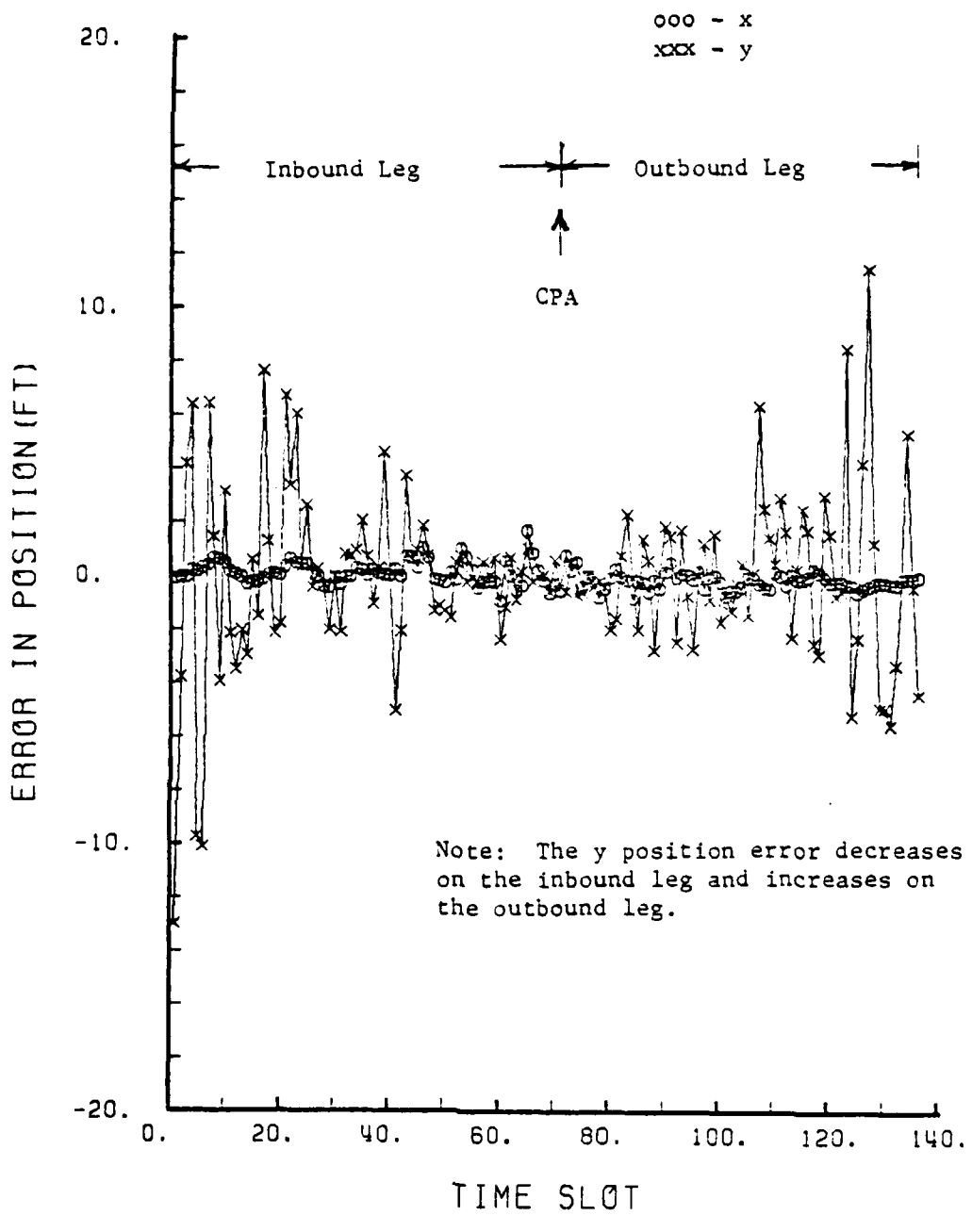


Figure 6. Error in Torpedo Position During A Straight Run in the Area of a Single Array

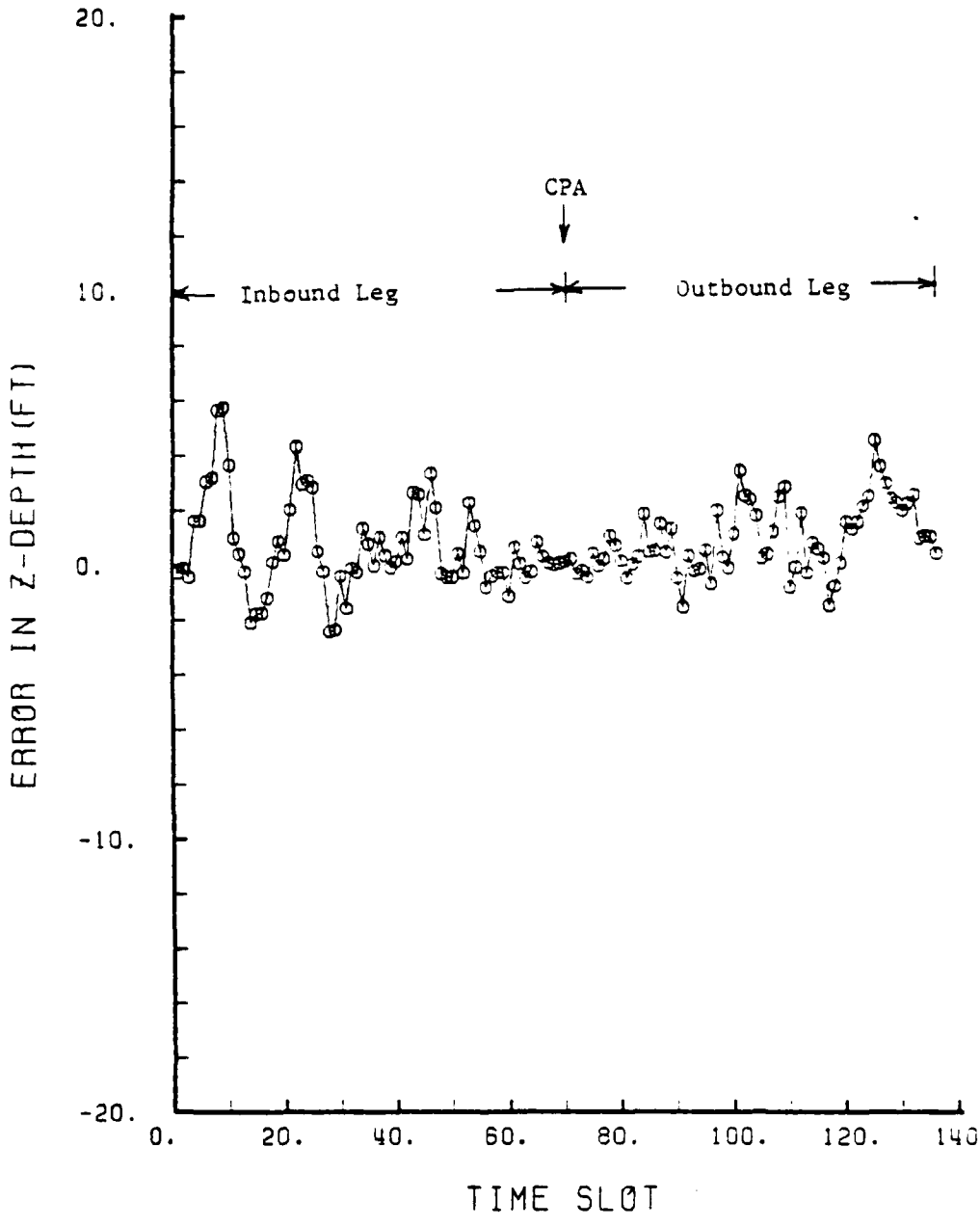


Figure 7. Error in Torpedo Deptn During a Straight Run in the Area of a Single Array

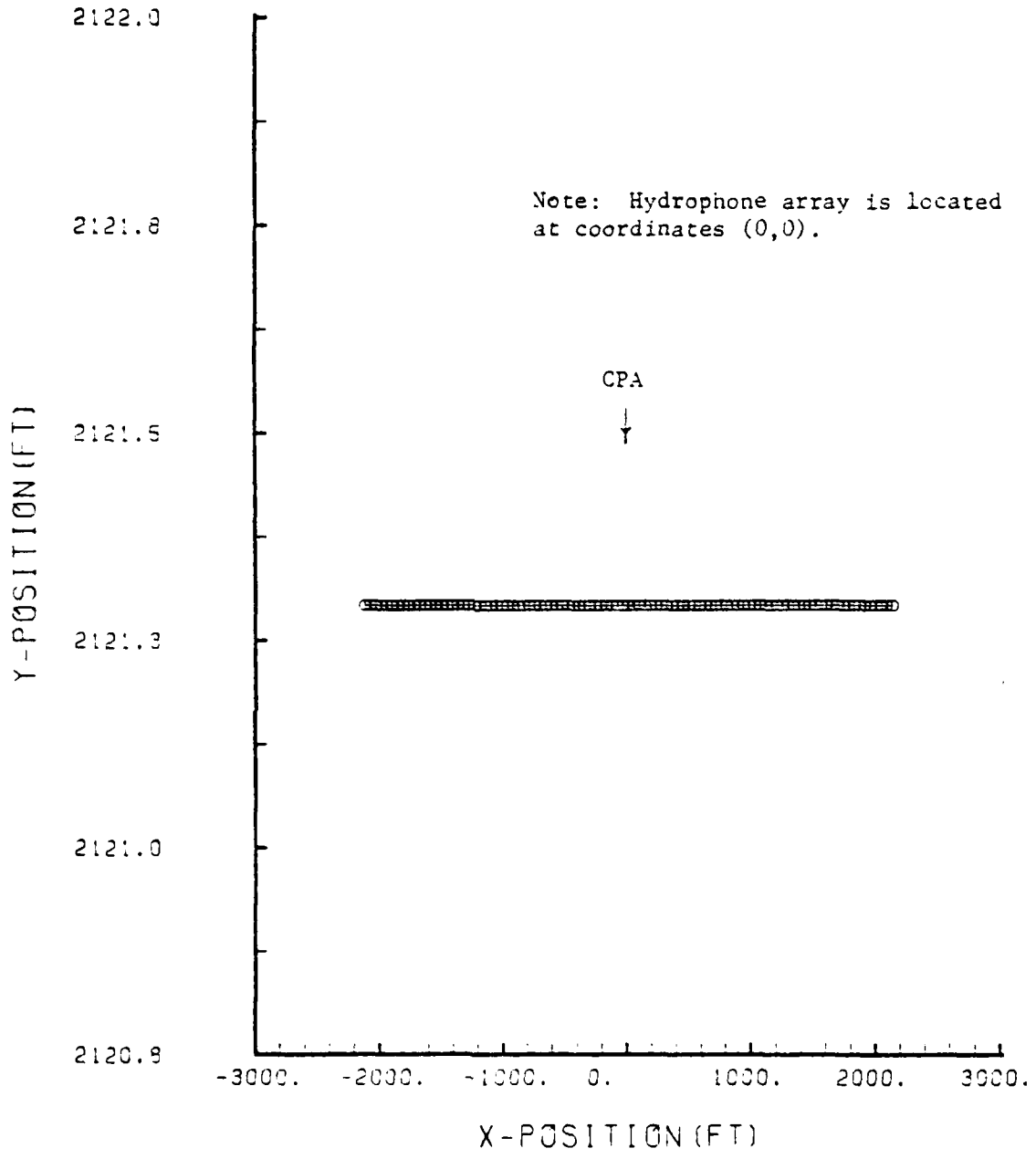


Figure 8. True Trajectory of the Torpedo During a Straight Run in the Area of a Single Array.

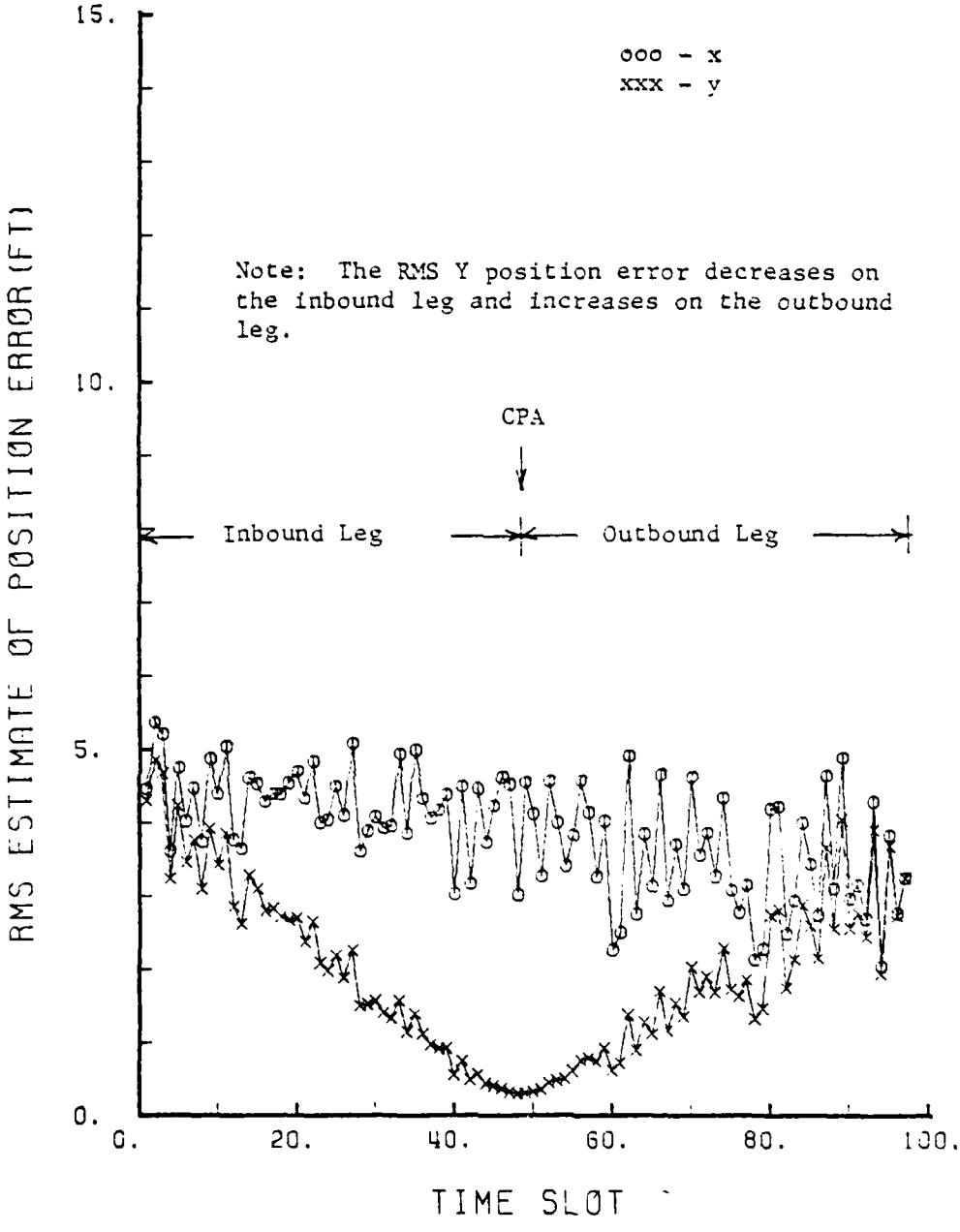


Figure 9. Filter Estimate of the RMS Position Error During a Straight Run in the Area of a Single Array

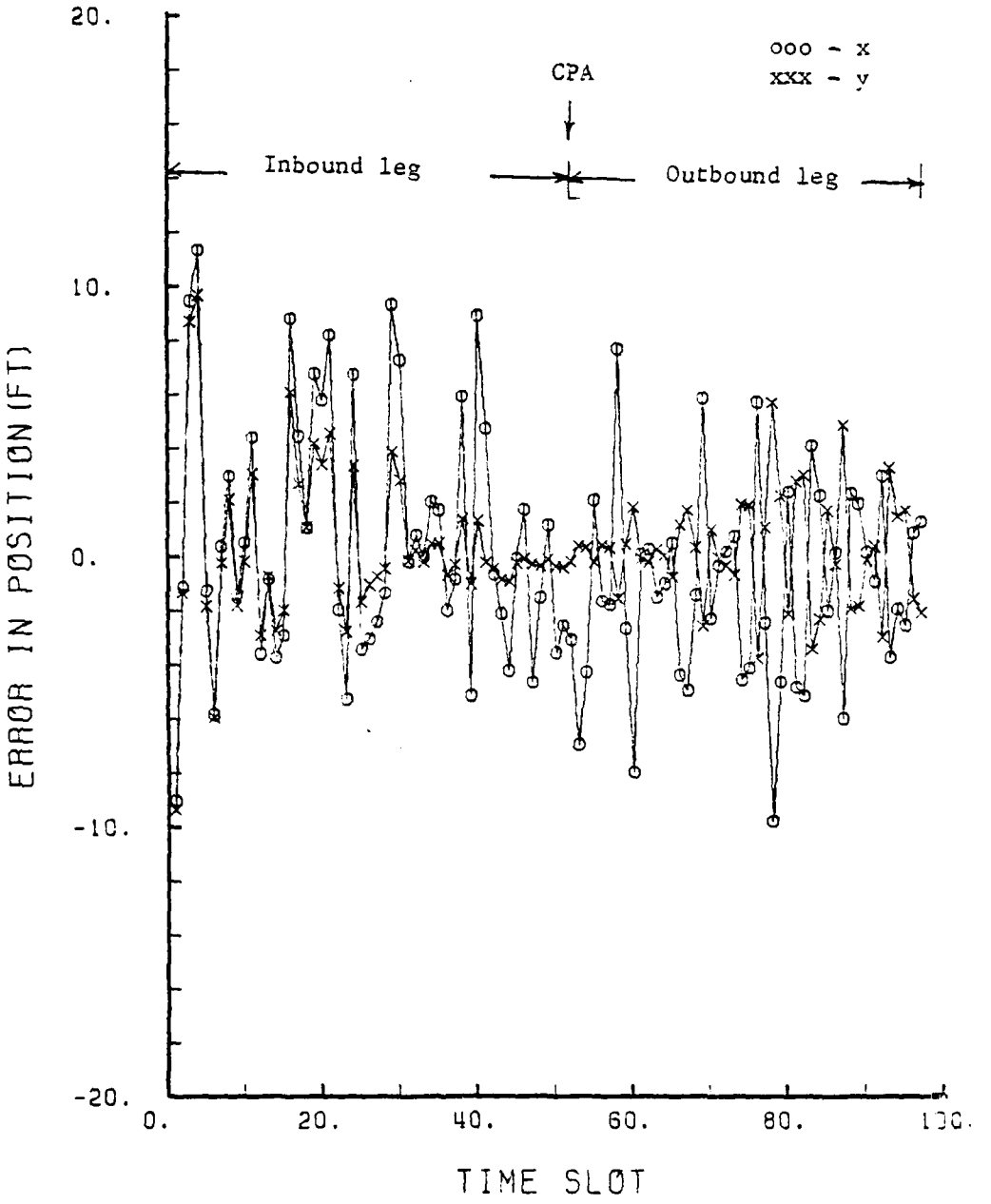


Figure 10. Error in Torpedo Position During a Straight Run in the Area of a Single Array

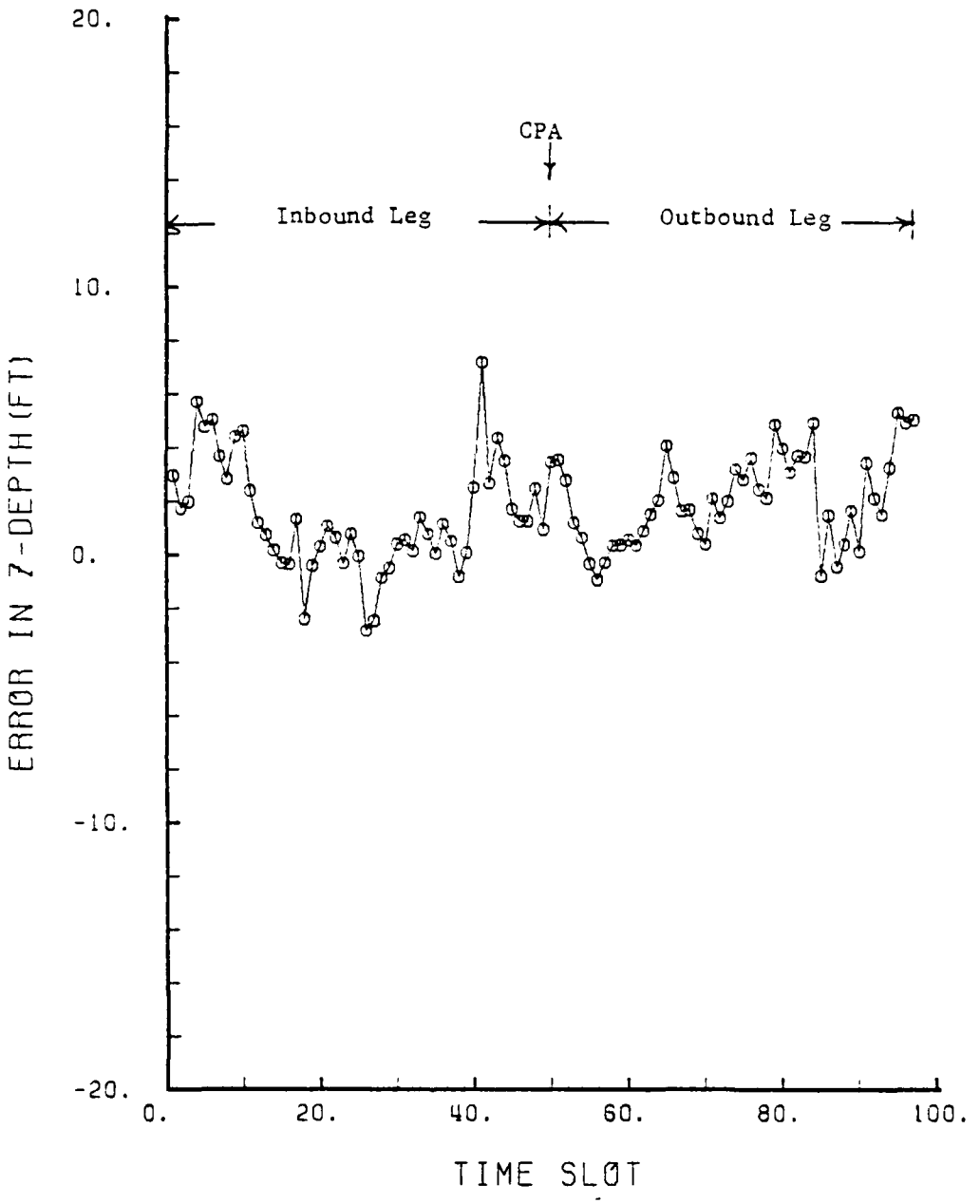


Figure 11. Error in Torpedo Depth During a Straight Run in the Area of a Single Array

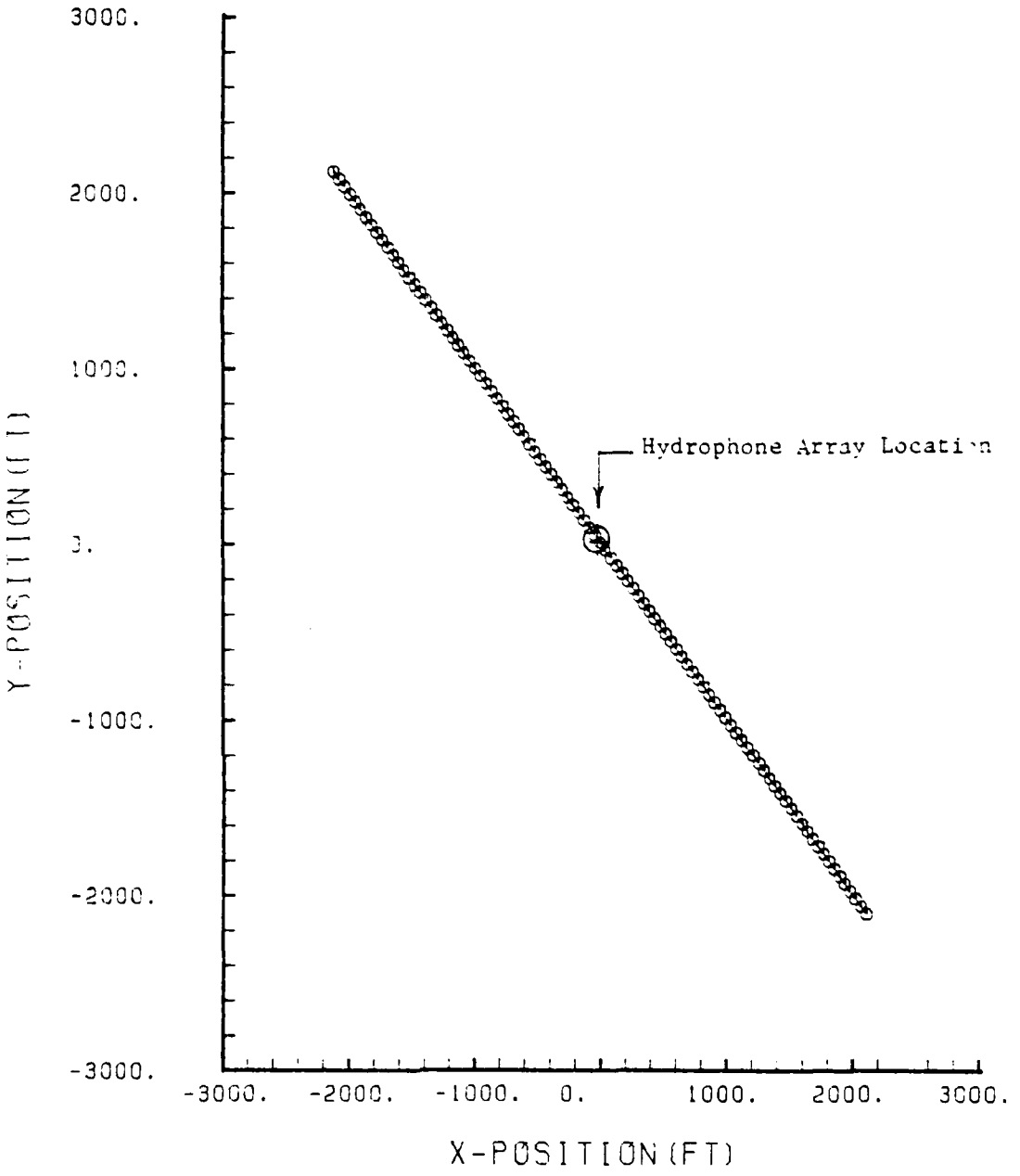


Figure 12. True Trajectory of the Torpedo During a Straight Run in the Area of a Single Array

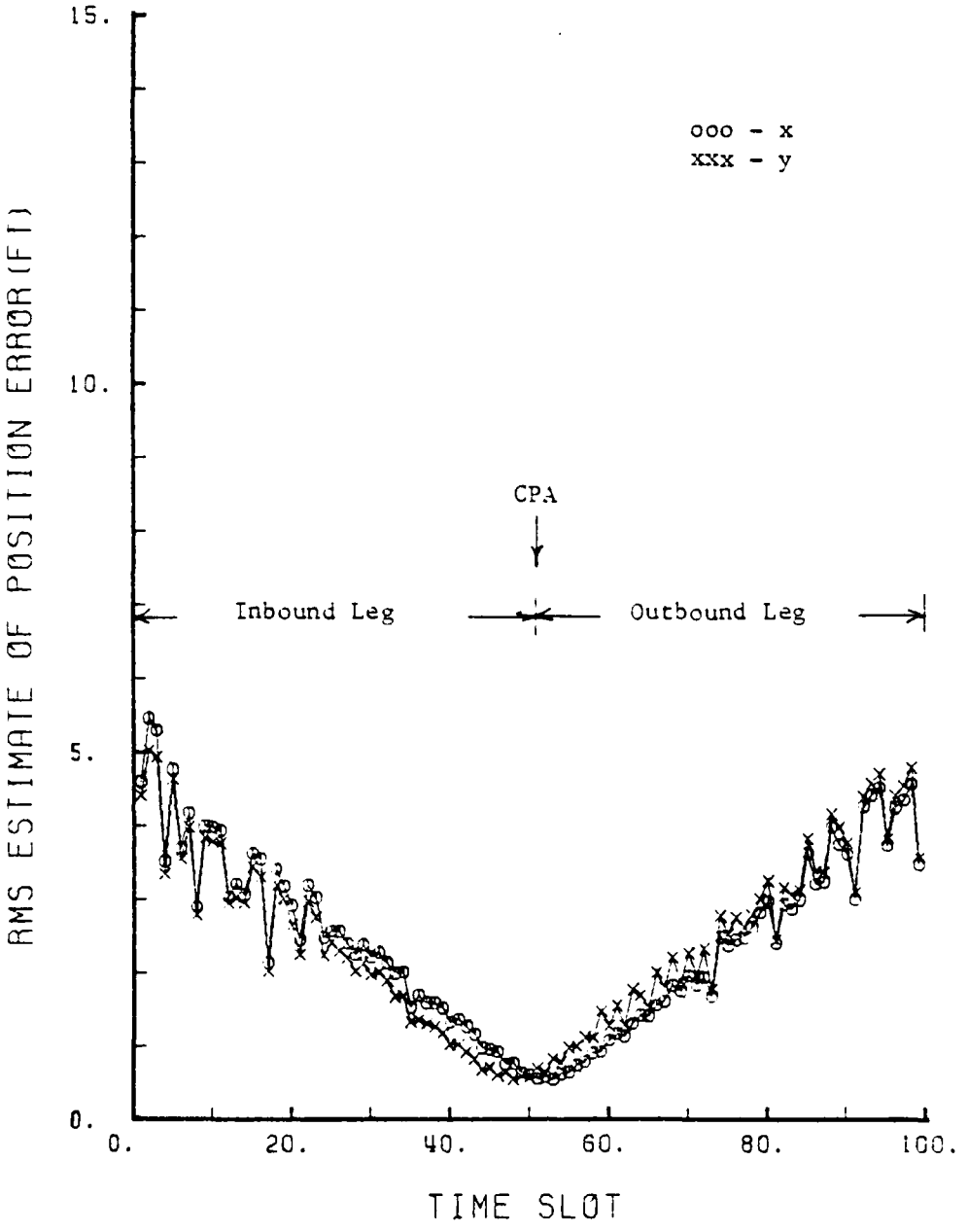


Figure 13. Filter Estimate of RMS Position Position Error During a Single Run in the Area of Single Array

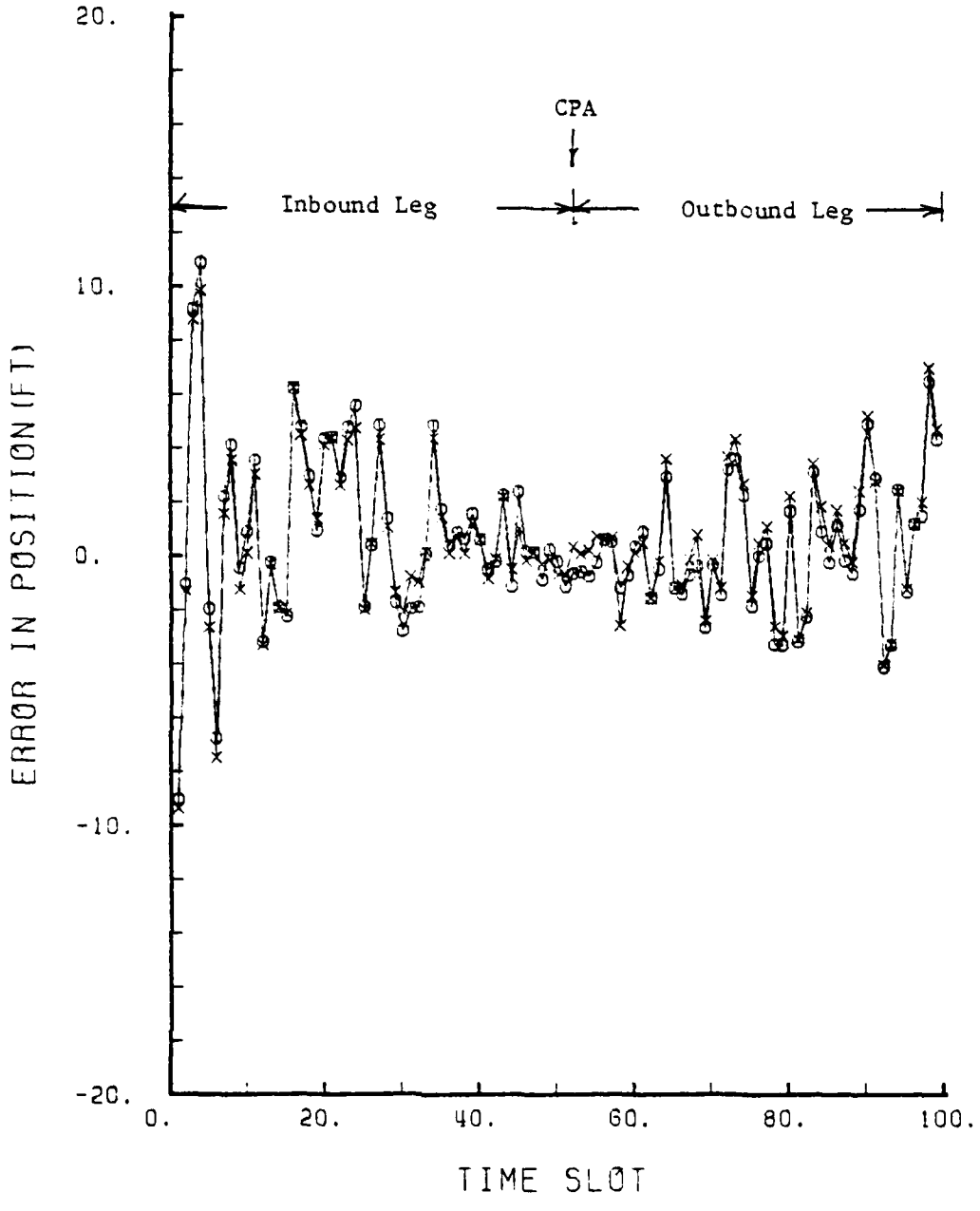


Figure 14. Error in Torpedo Position During a Straight Run in the Area of a Single Array

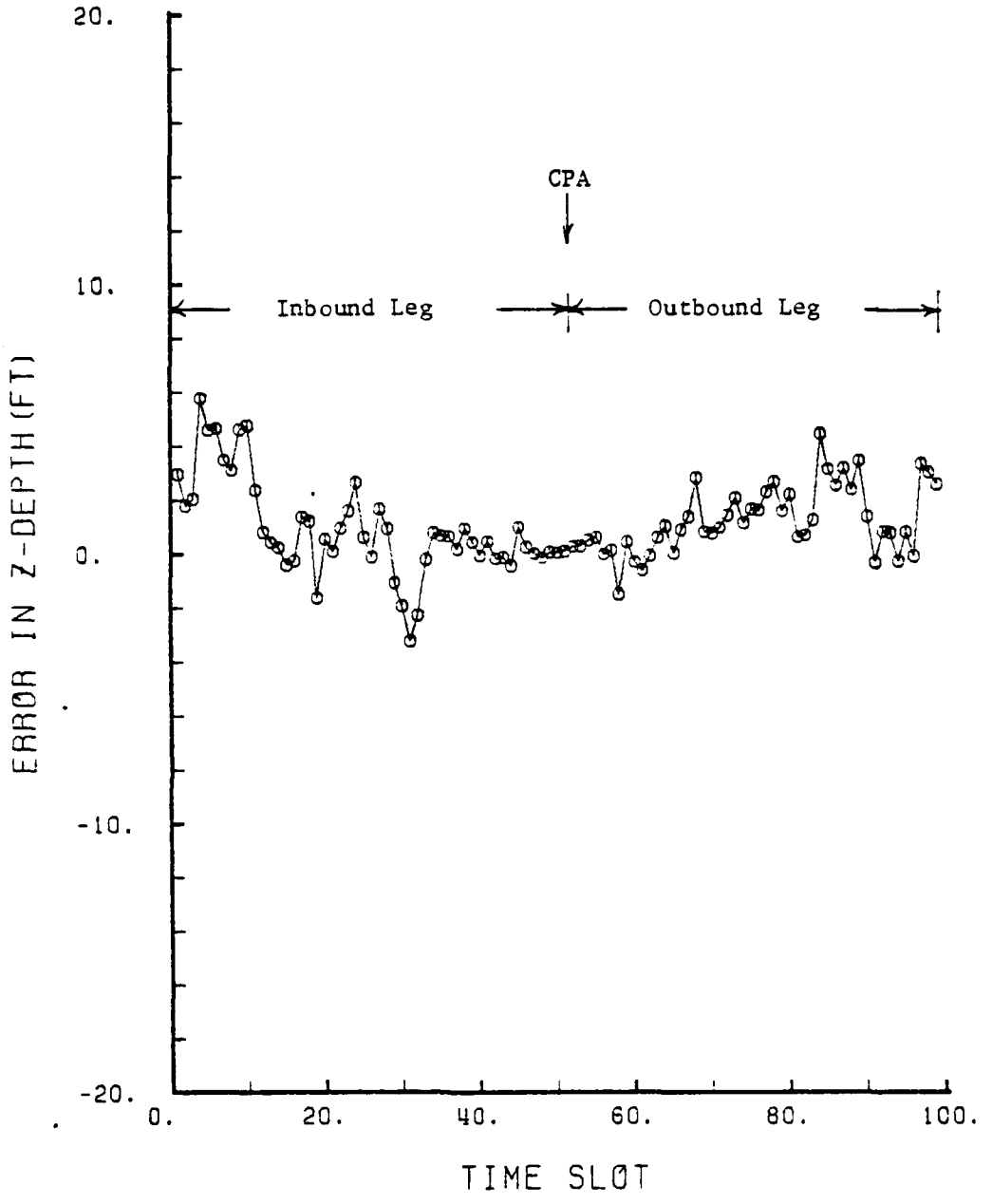


Figure 15. Error in Torpedo Depth During a Straight Run in the Area of a Single Array

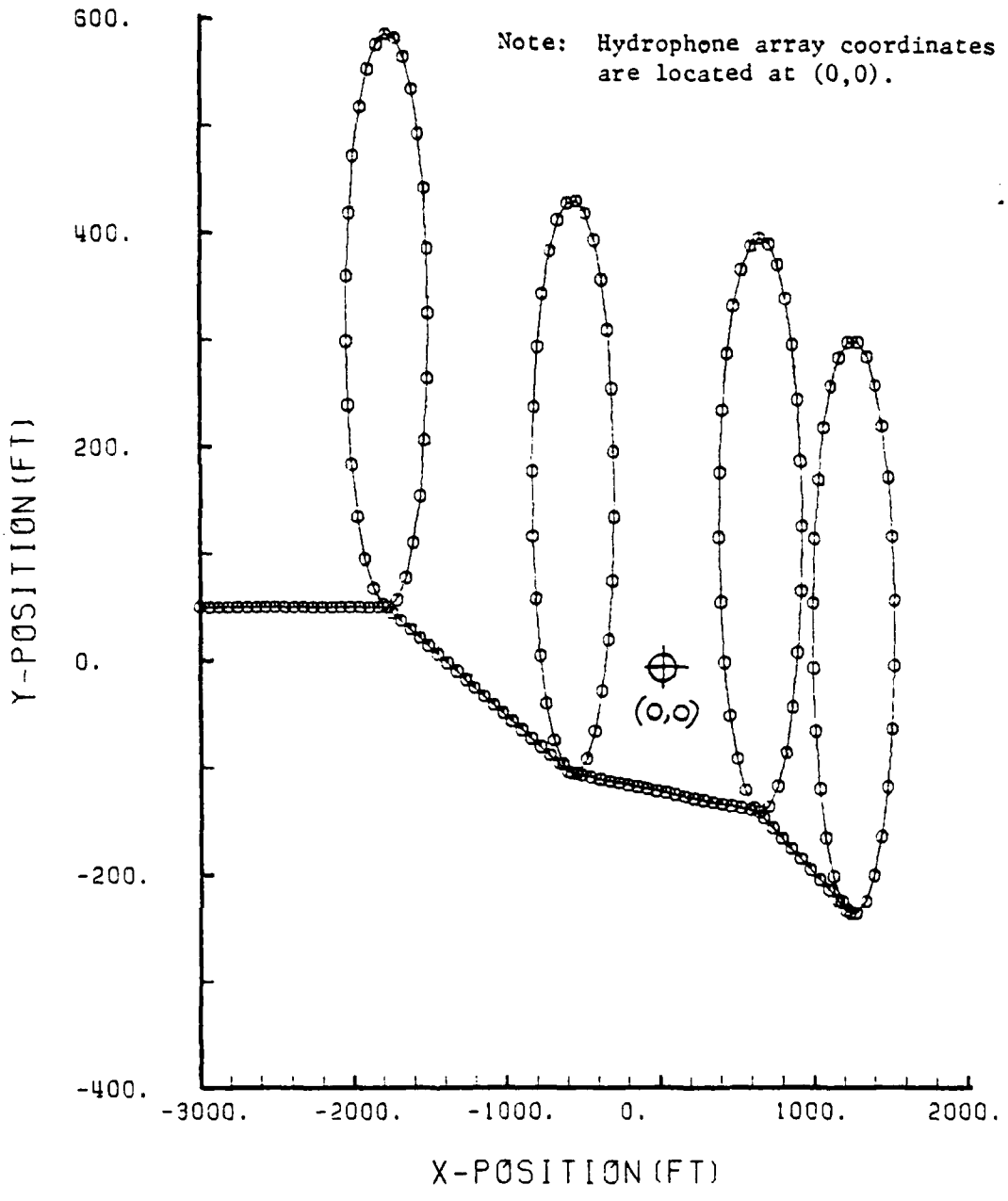


Figure 16. True Trajectory of the Torpedo in the Area of a Single Array

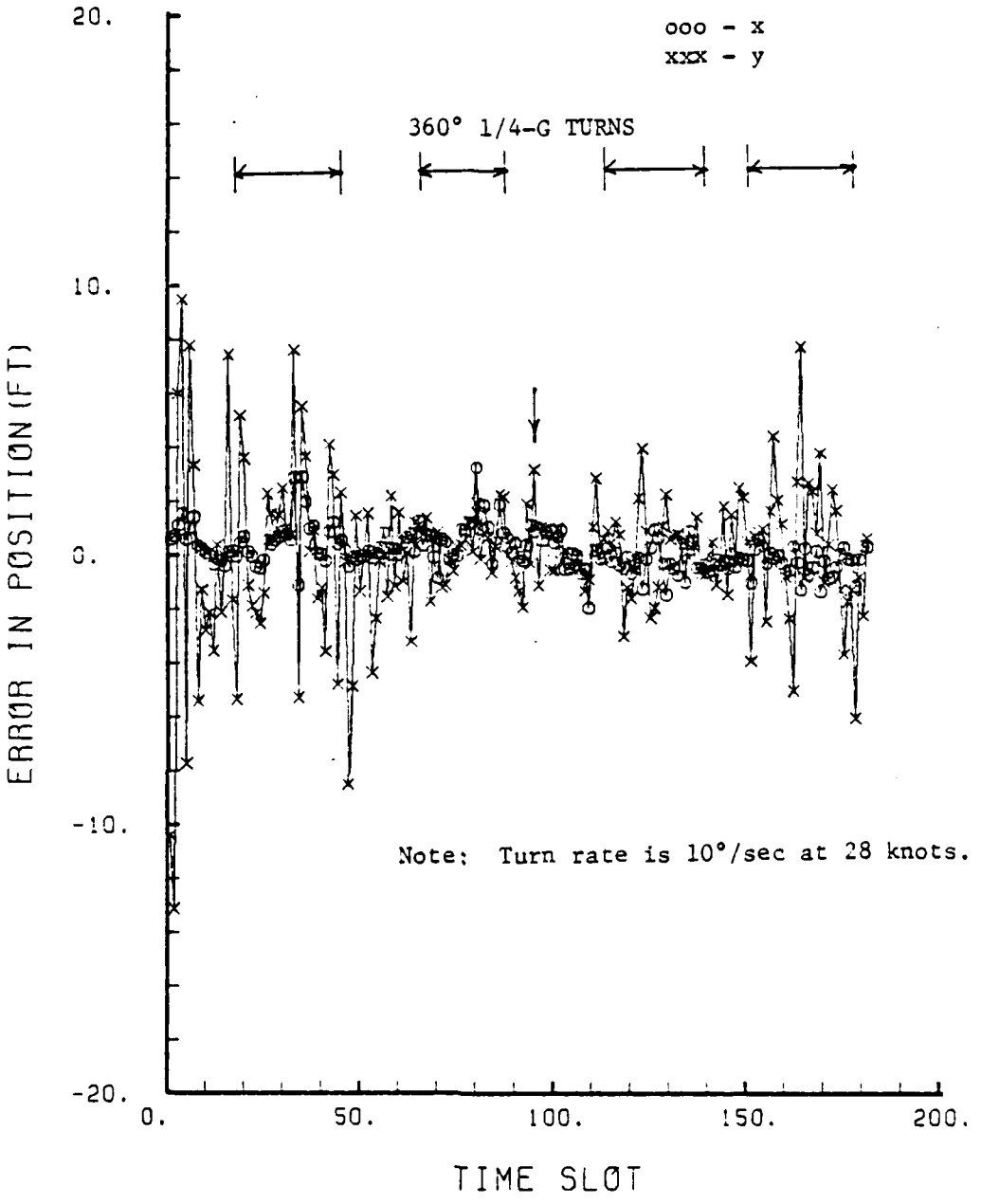


Figure 17. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

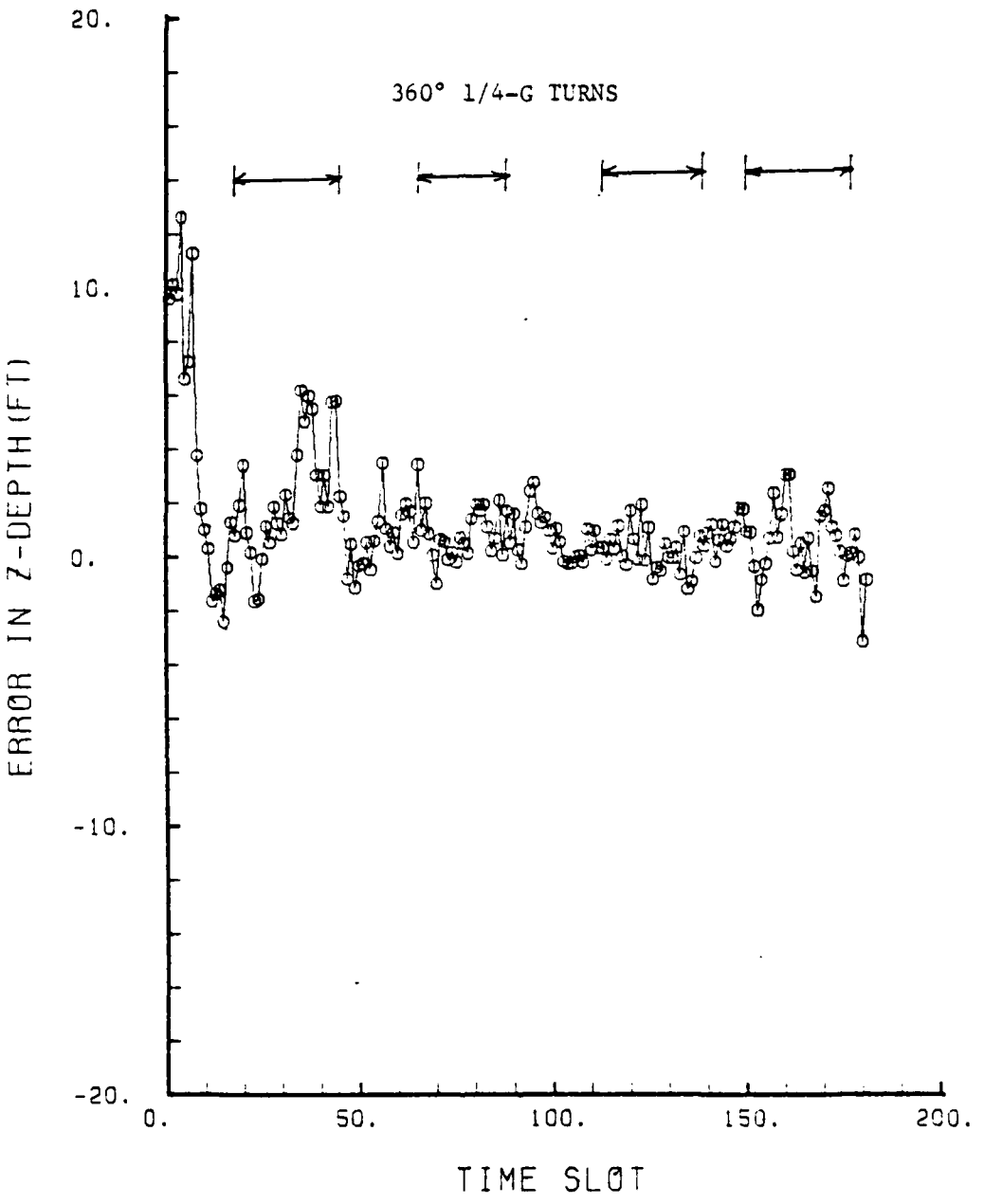


Figure 18. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

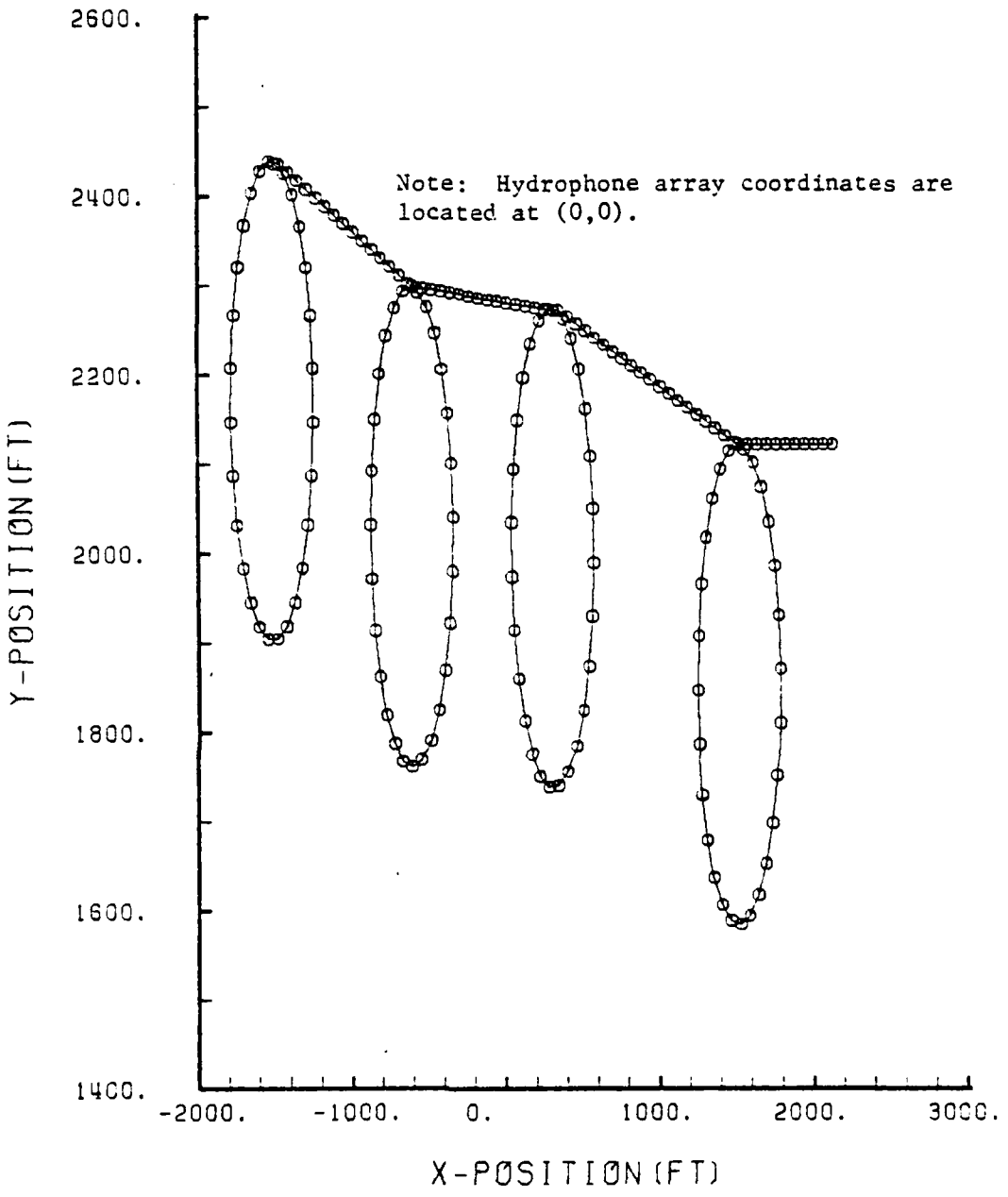


Figure 19. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

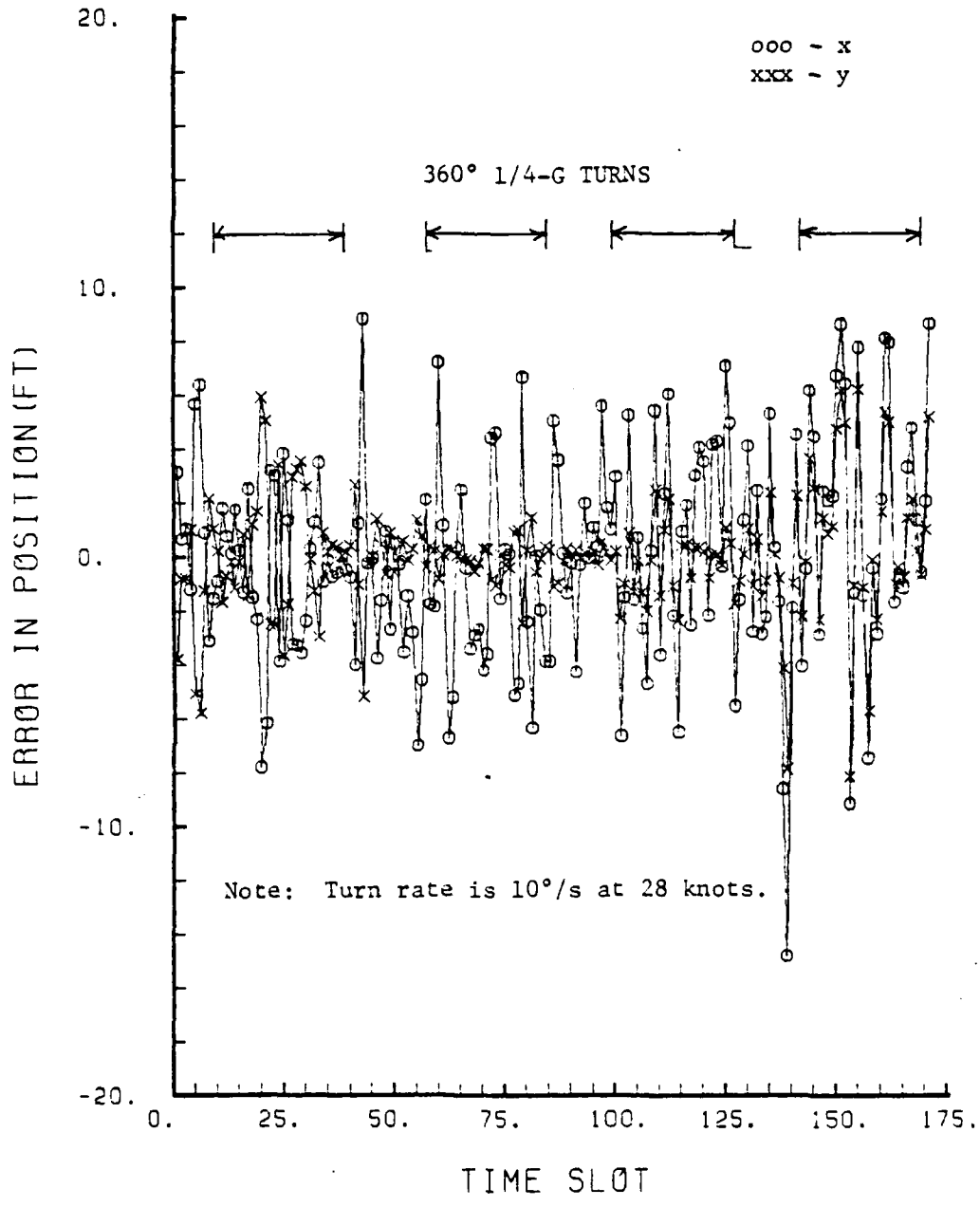


Figure 20. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

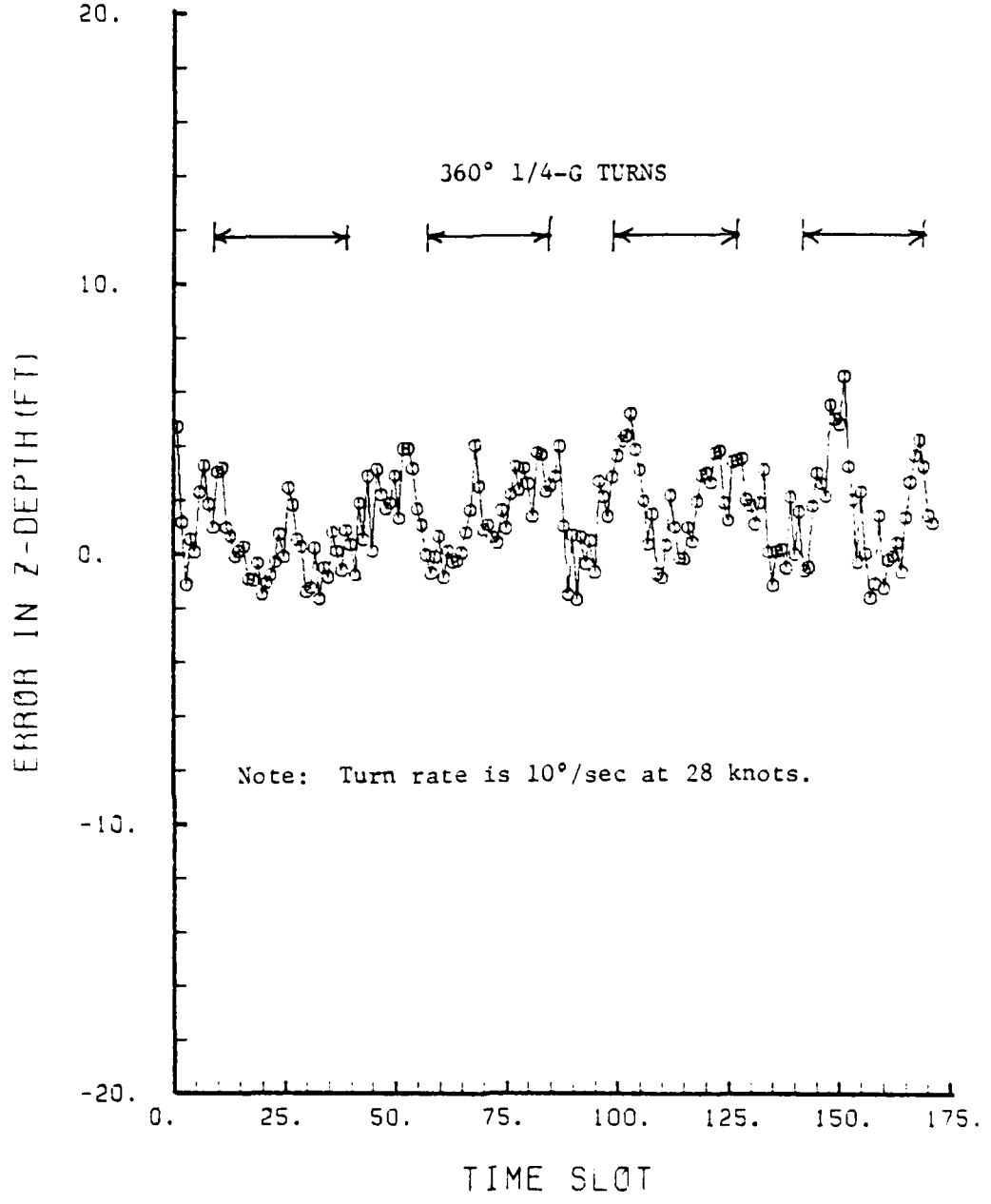


Figure 21. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

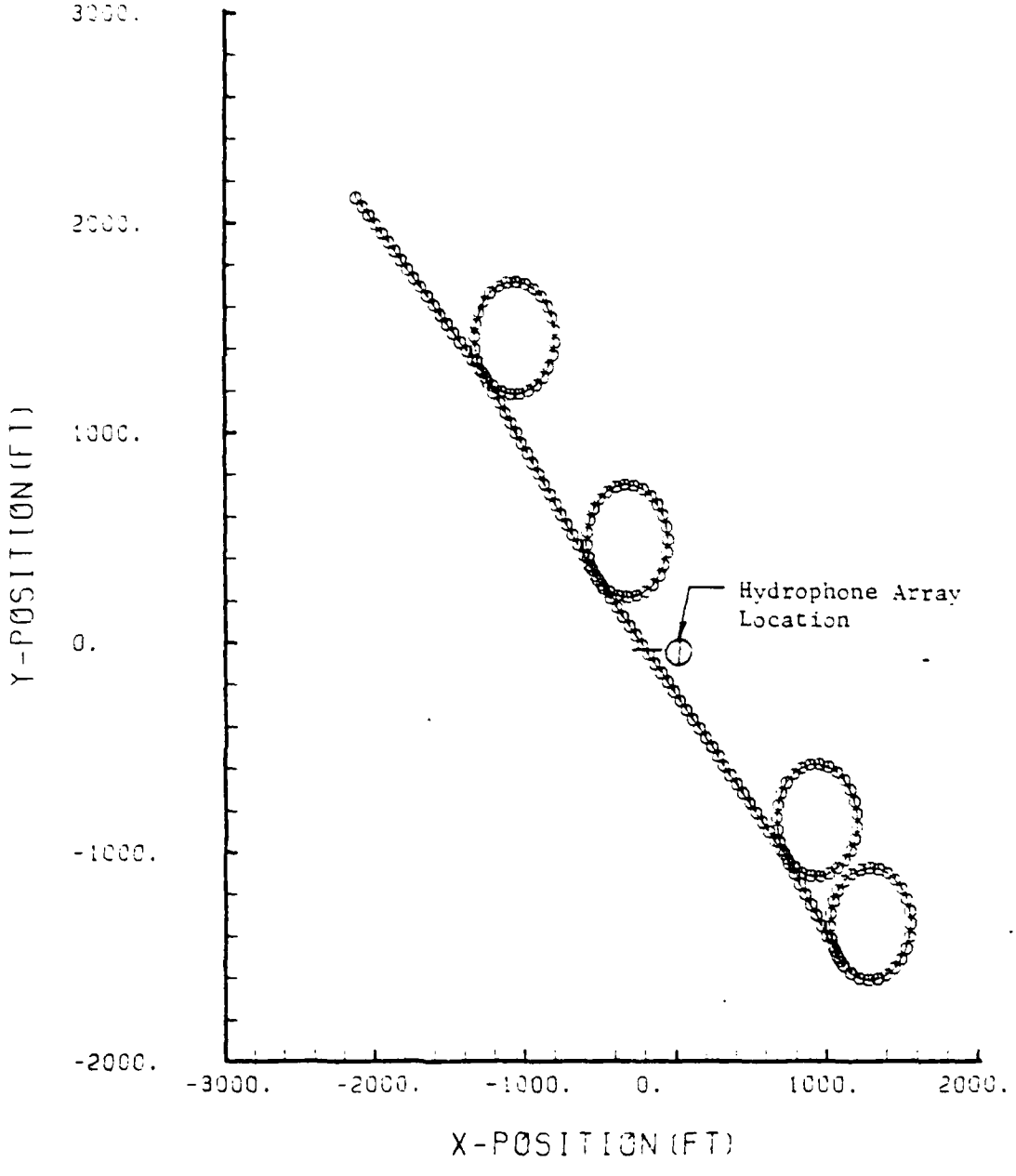


Figure 22. True Trajectory of the Torpedo During a Maneuvering Run in the Area of A Single Array

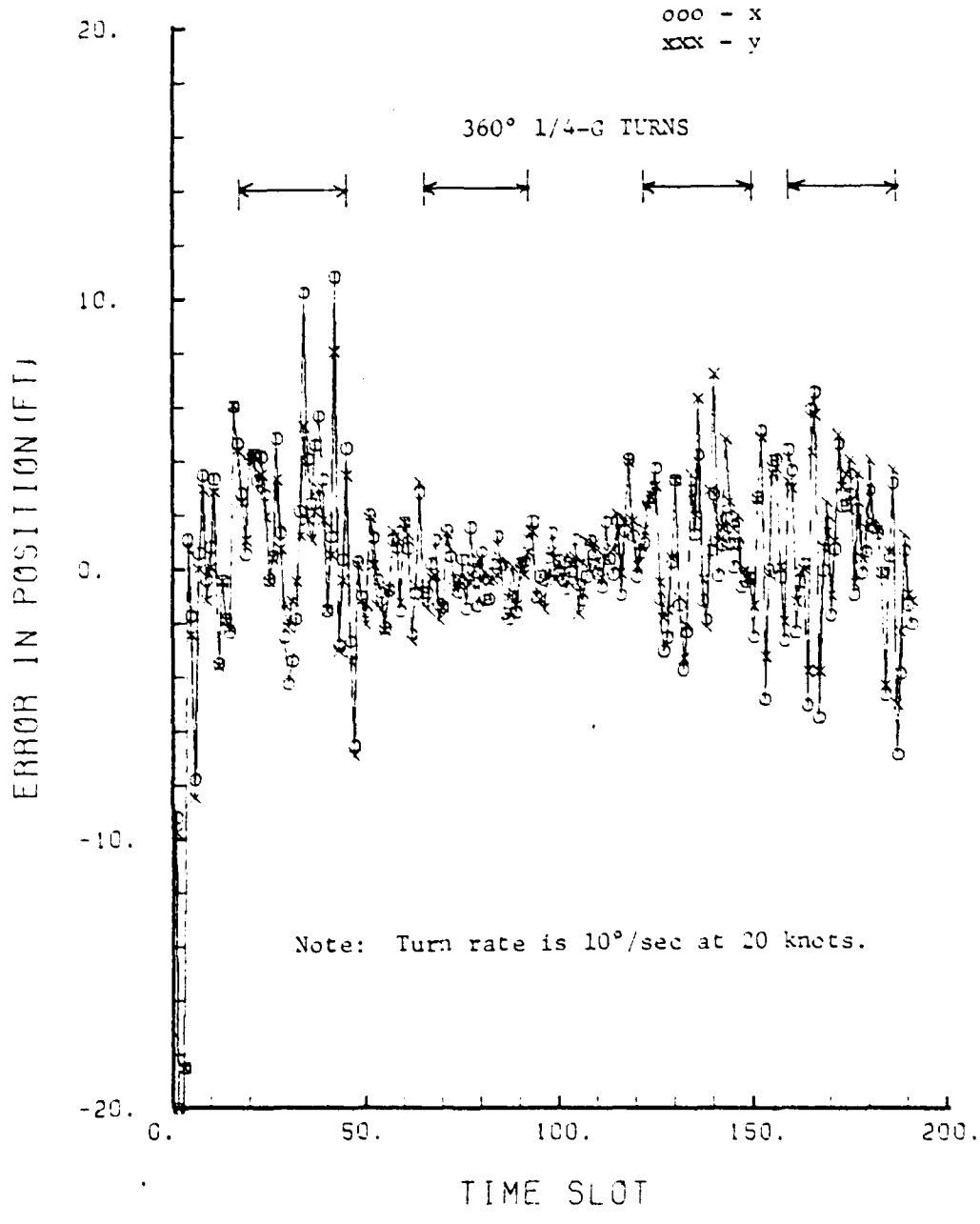


Figure 23. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

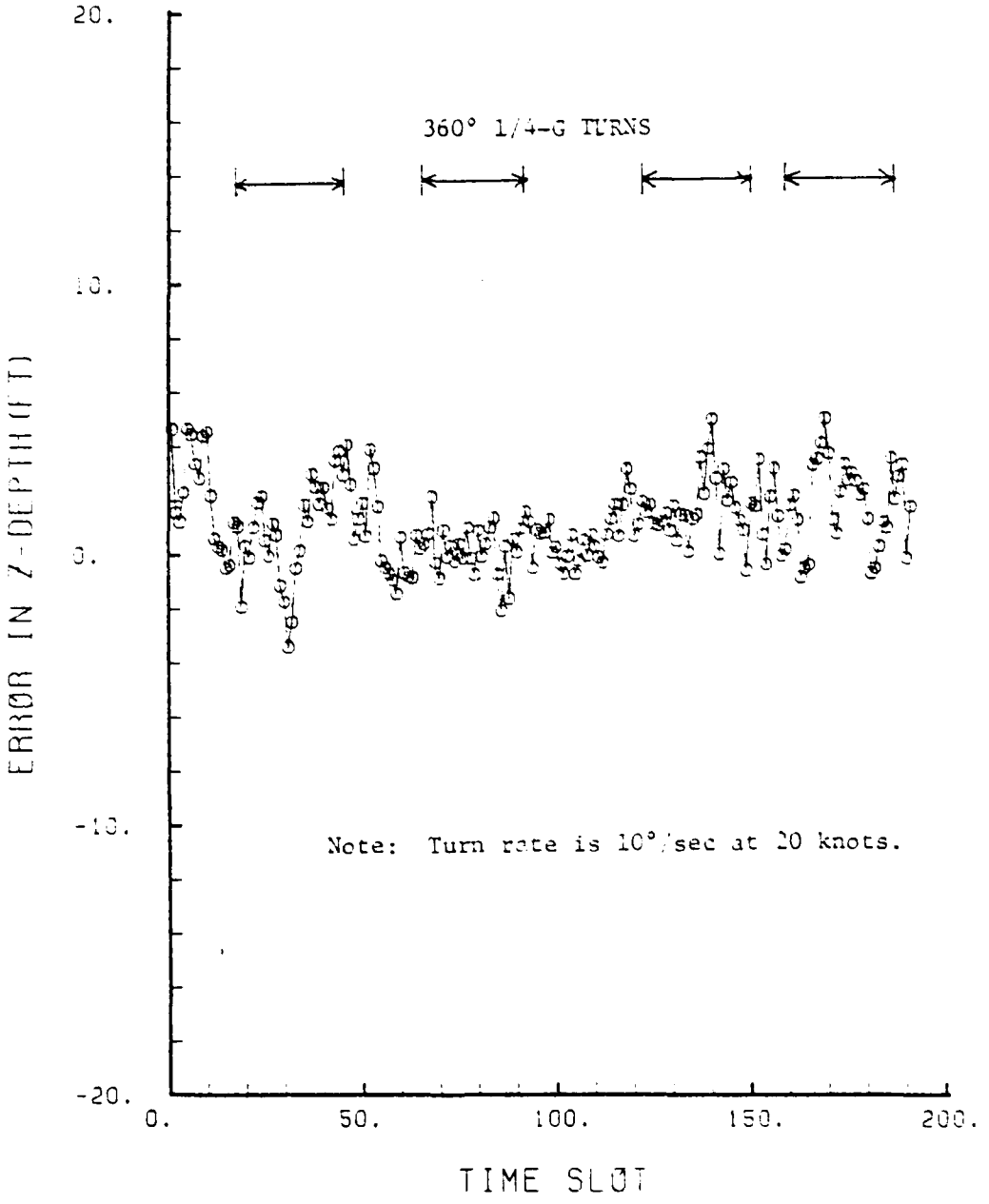


Figure 24. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

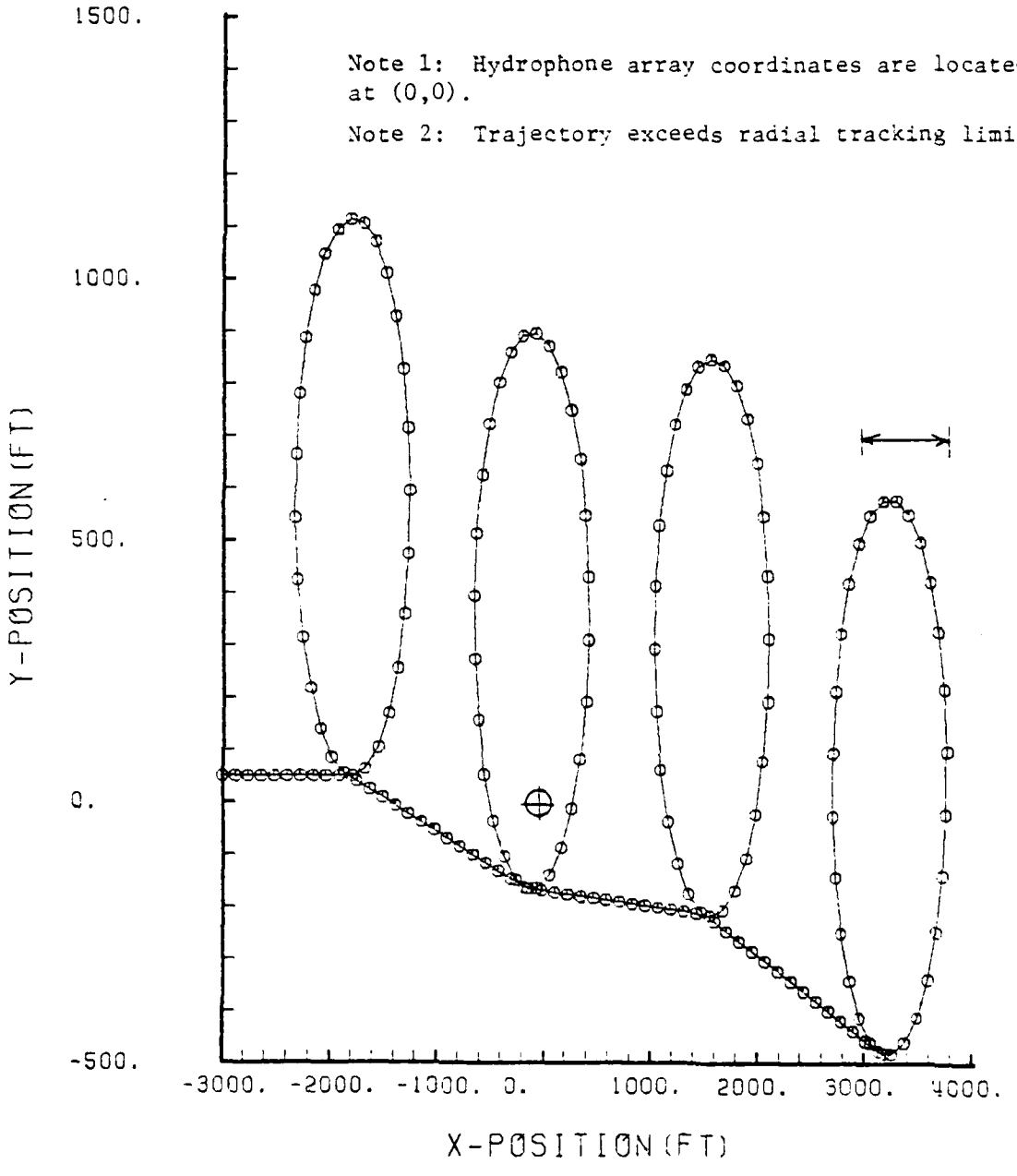


Figure 25. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

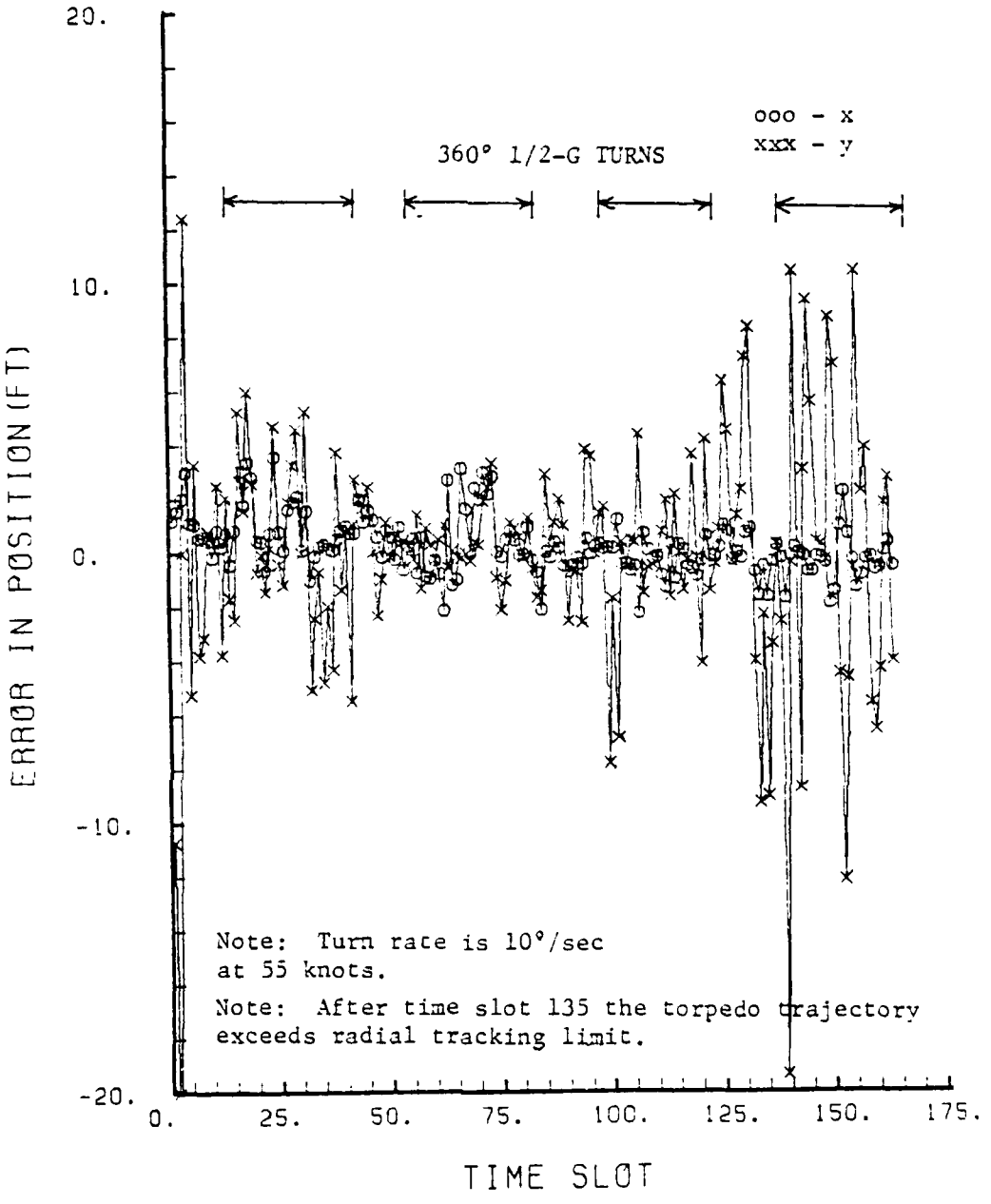


Figure 26. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

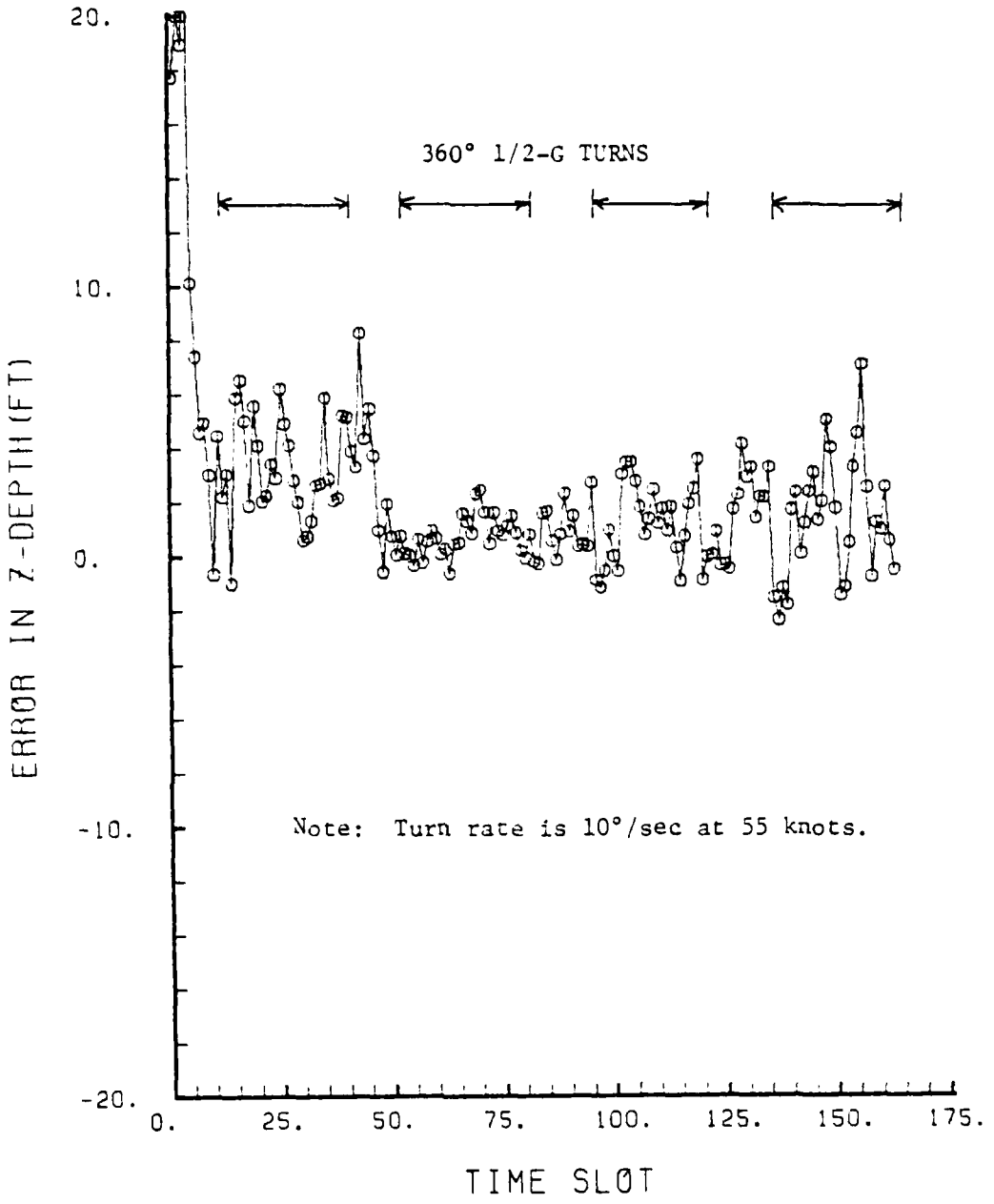


Figure 27. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

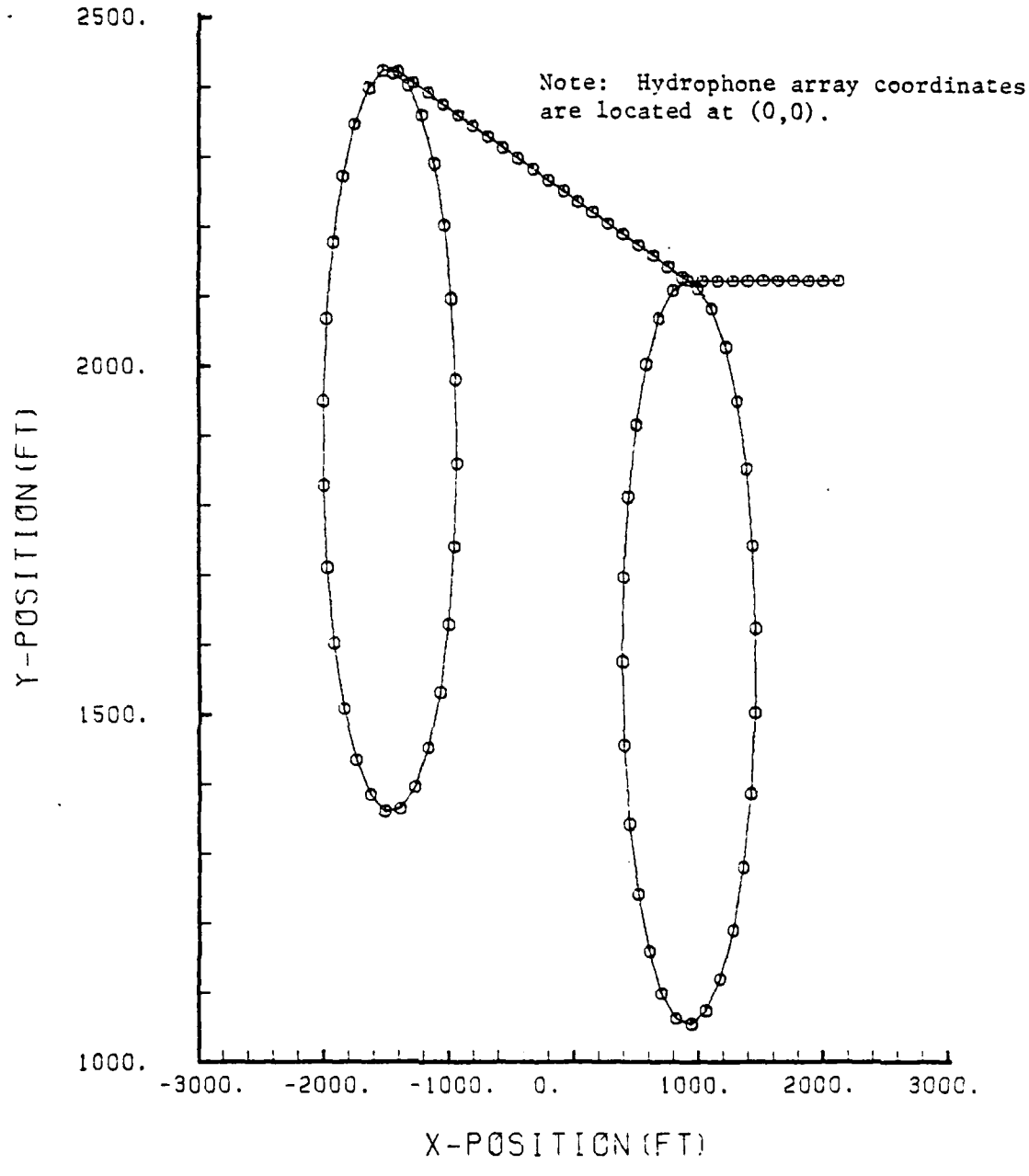


Figure 28. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

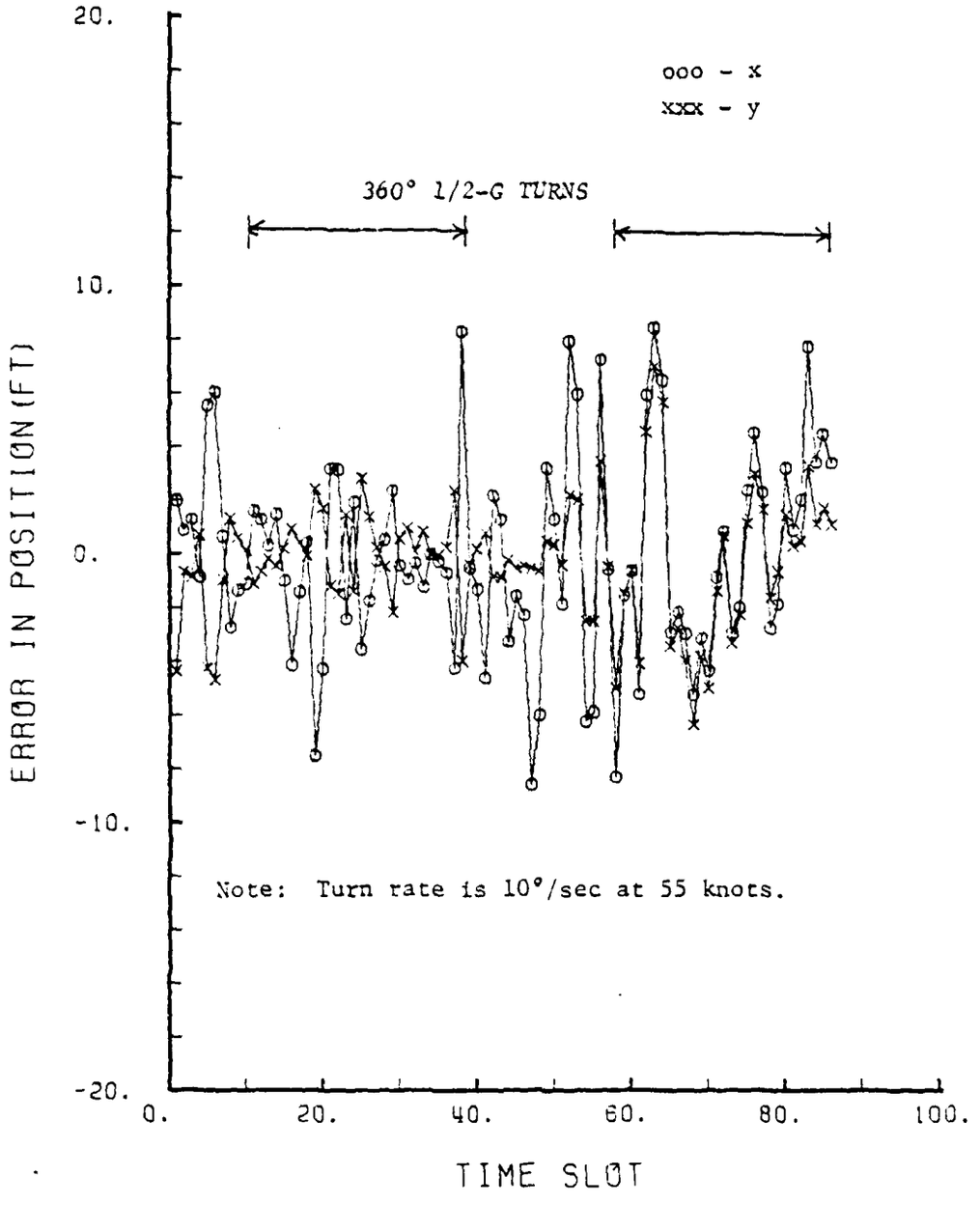


Figure 29. Error in Torpedo Position Duirng a Maneuvering Run in the Area of a Single Array

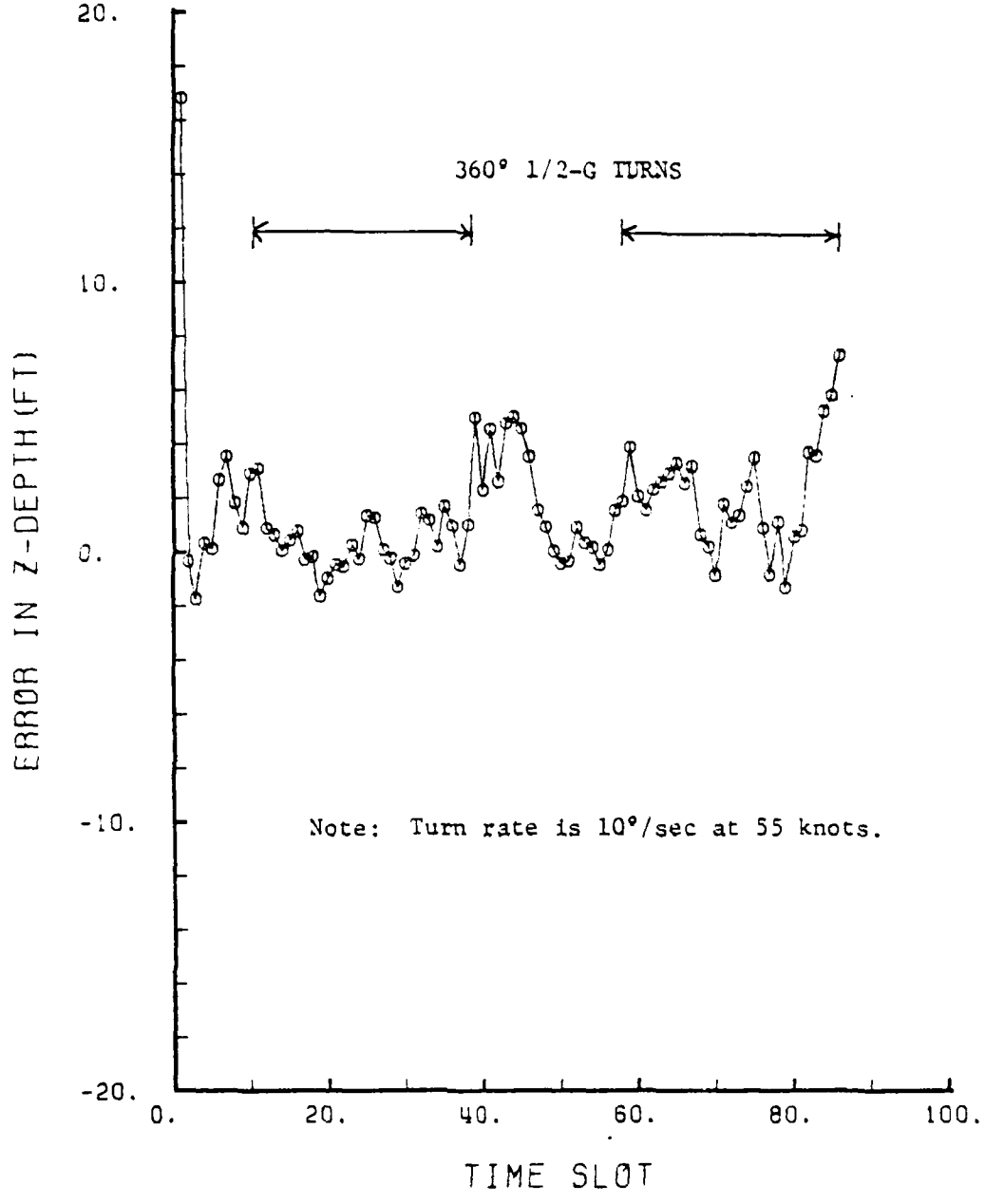


Figure 30. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

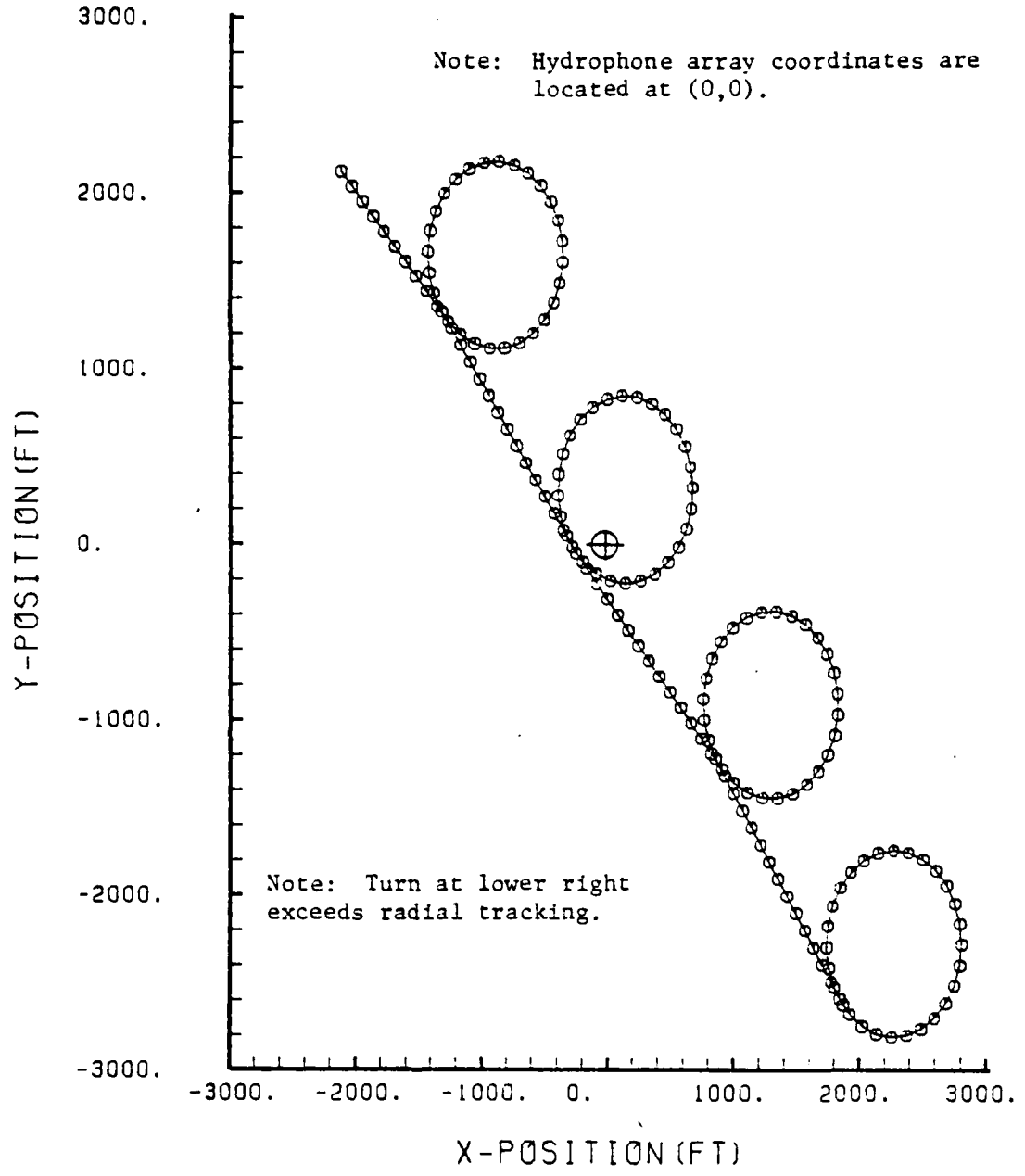


Figure 31. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

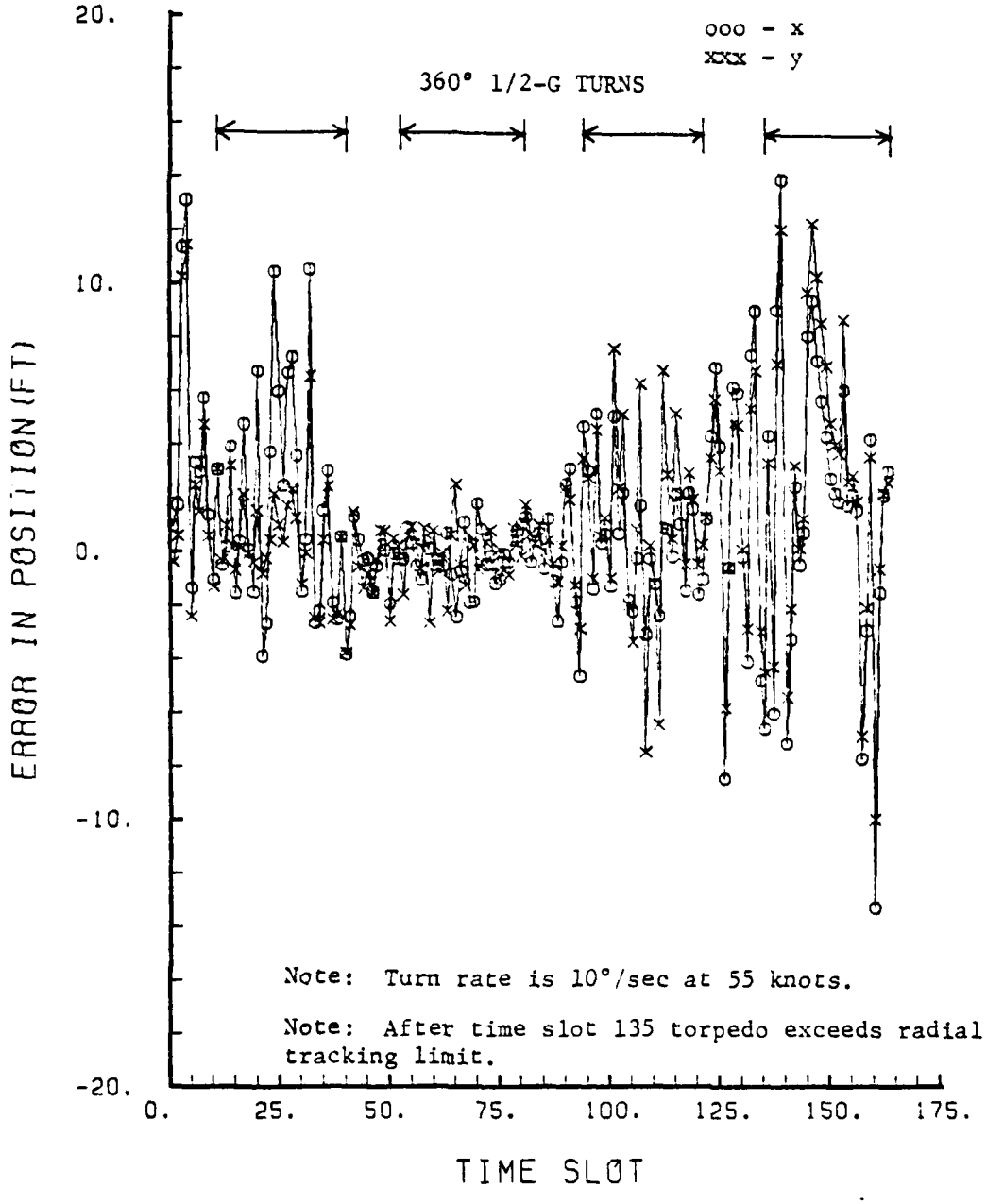


Figure 32. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

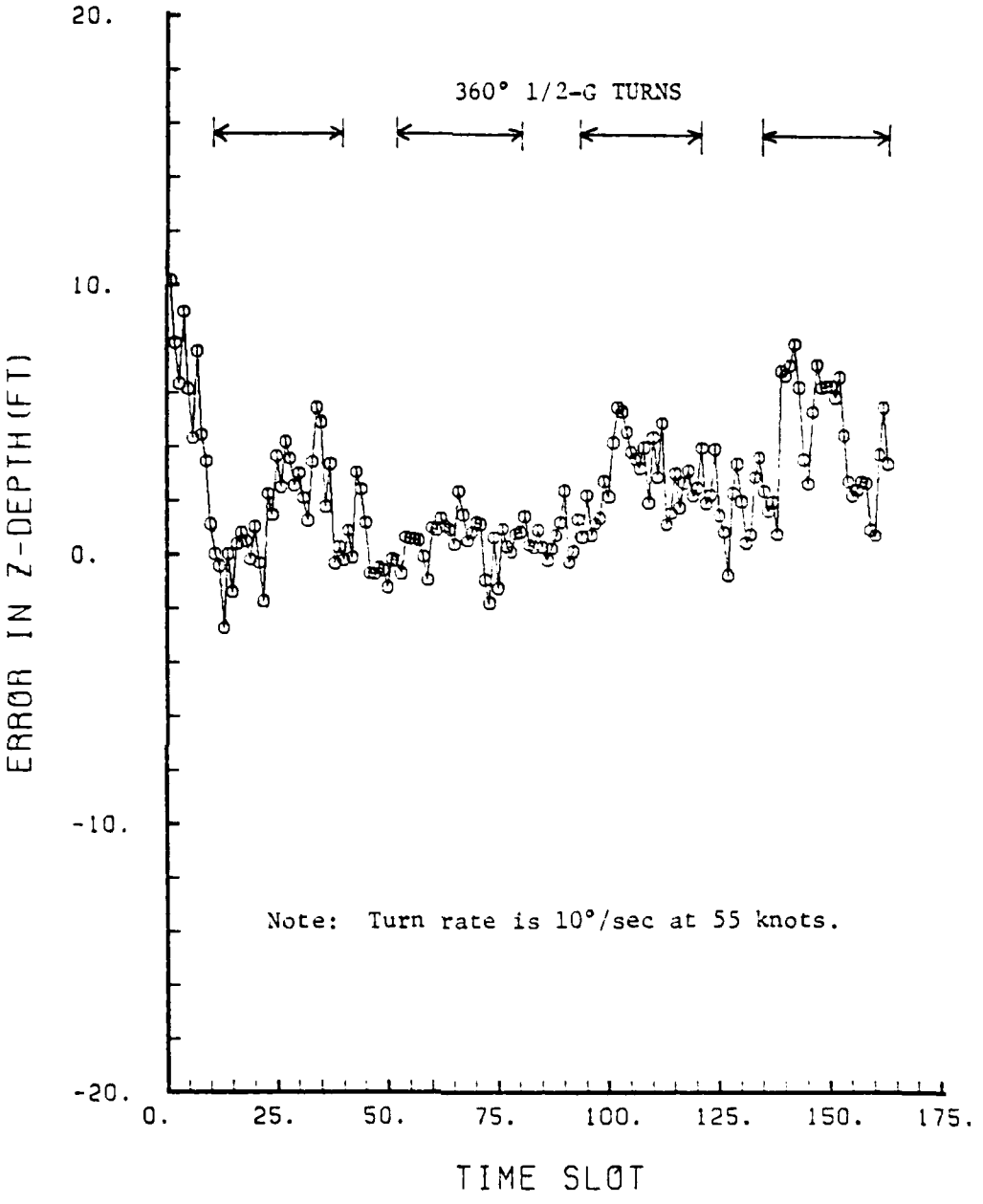


Figure 33. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

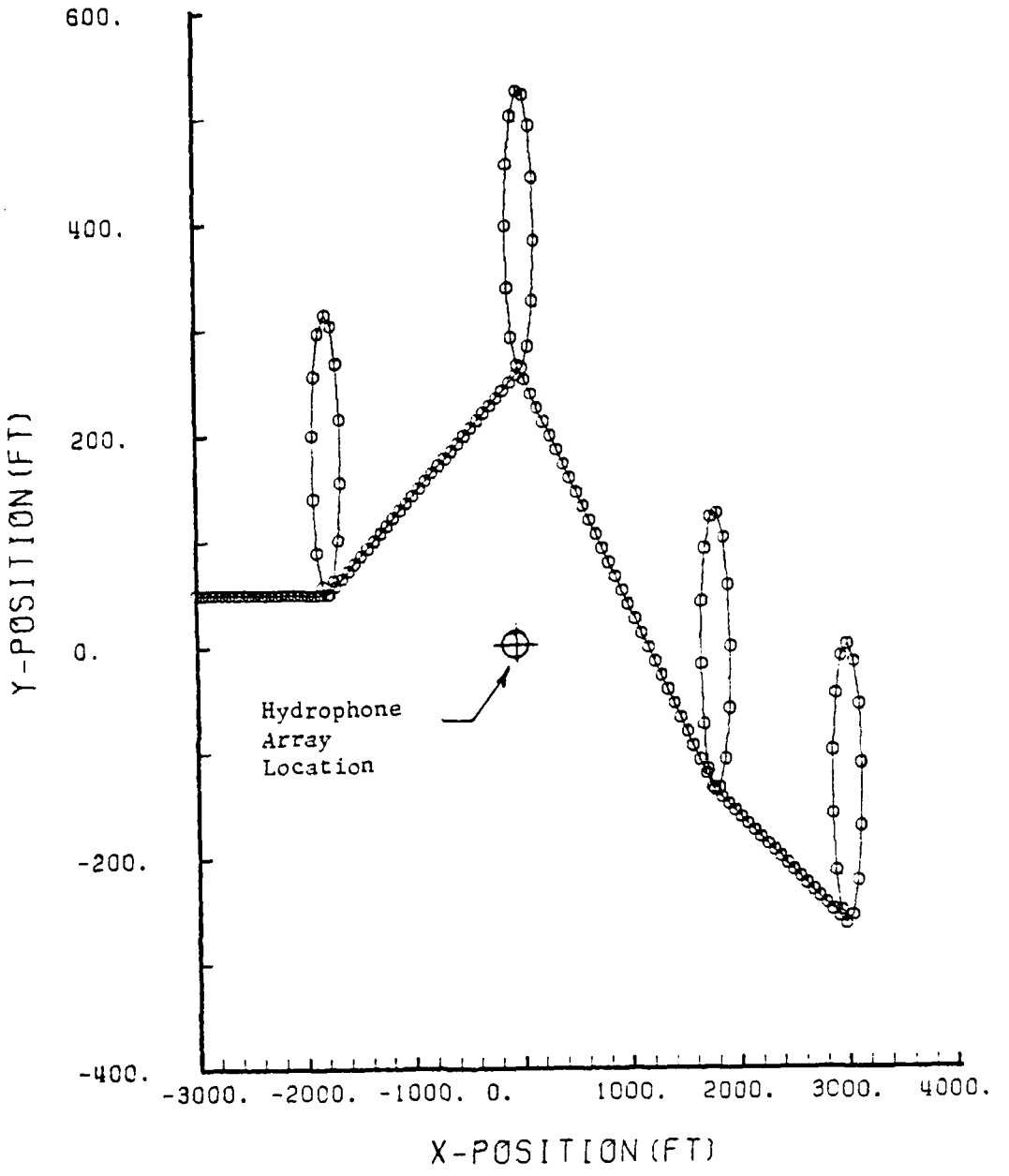


Figure 34. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

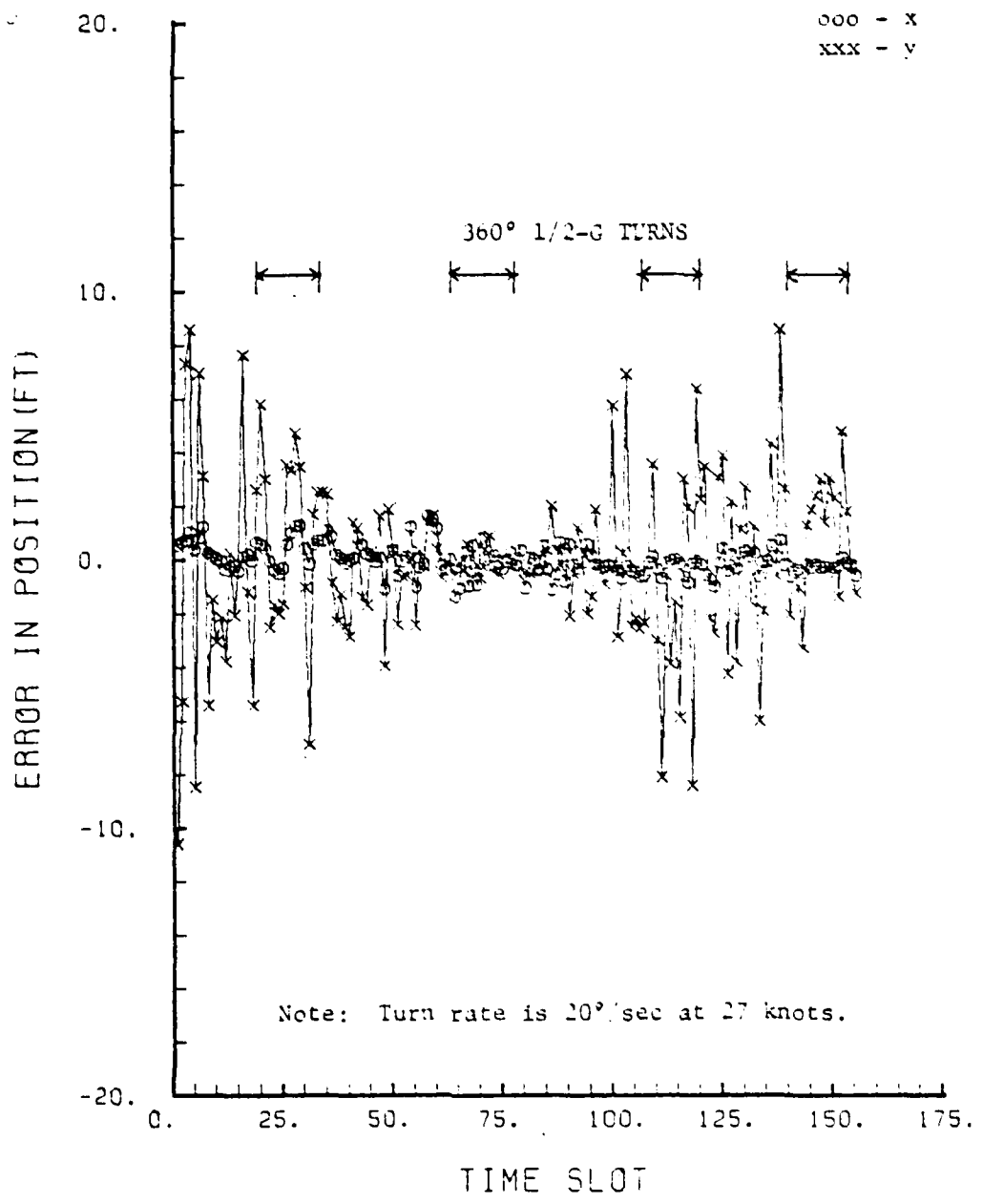


Figure 35. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

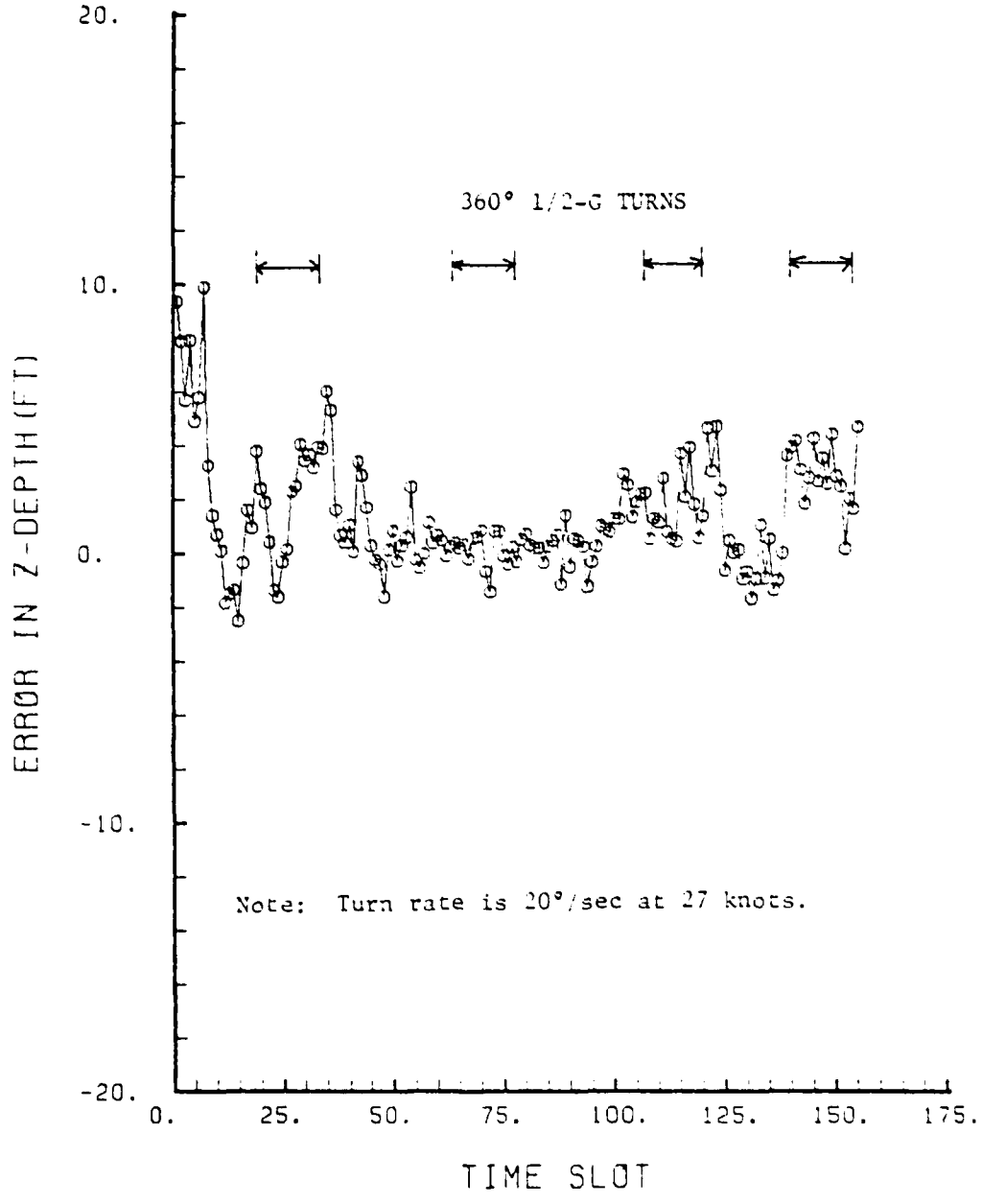


Figure 36. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

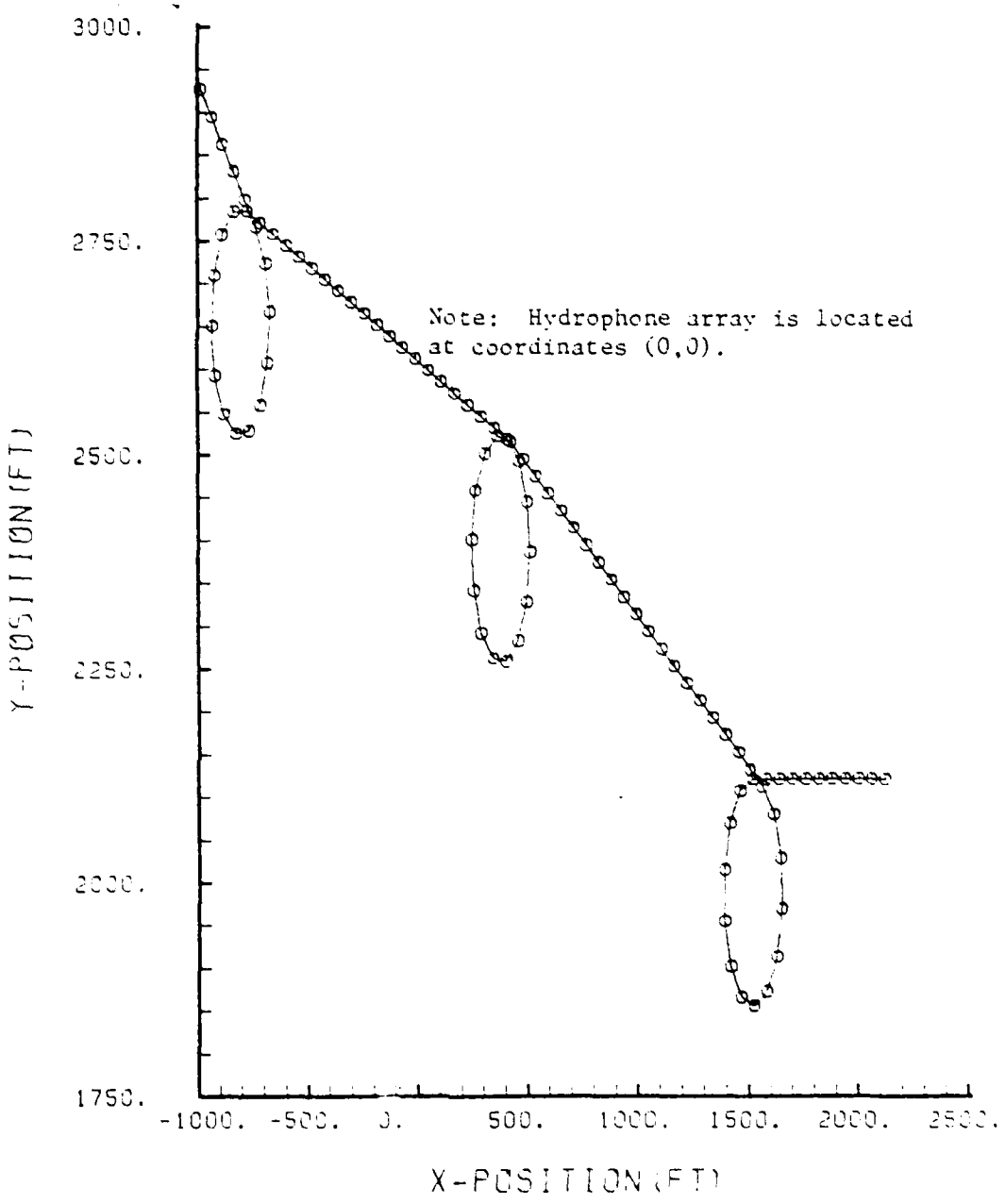


Figure 37. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

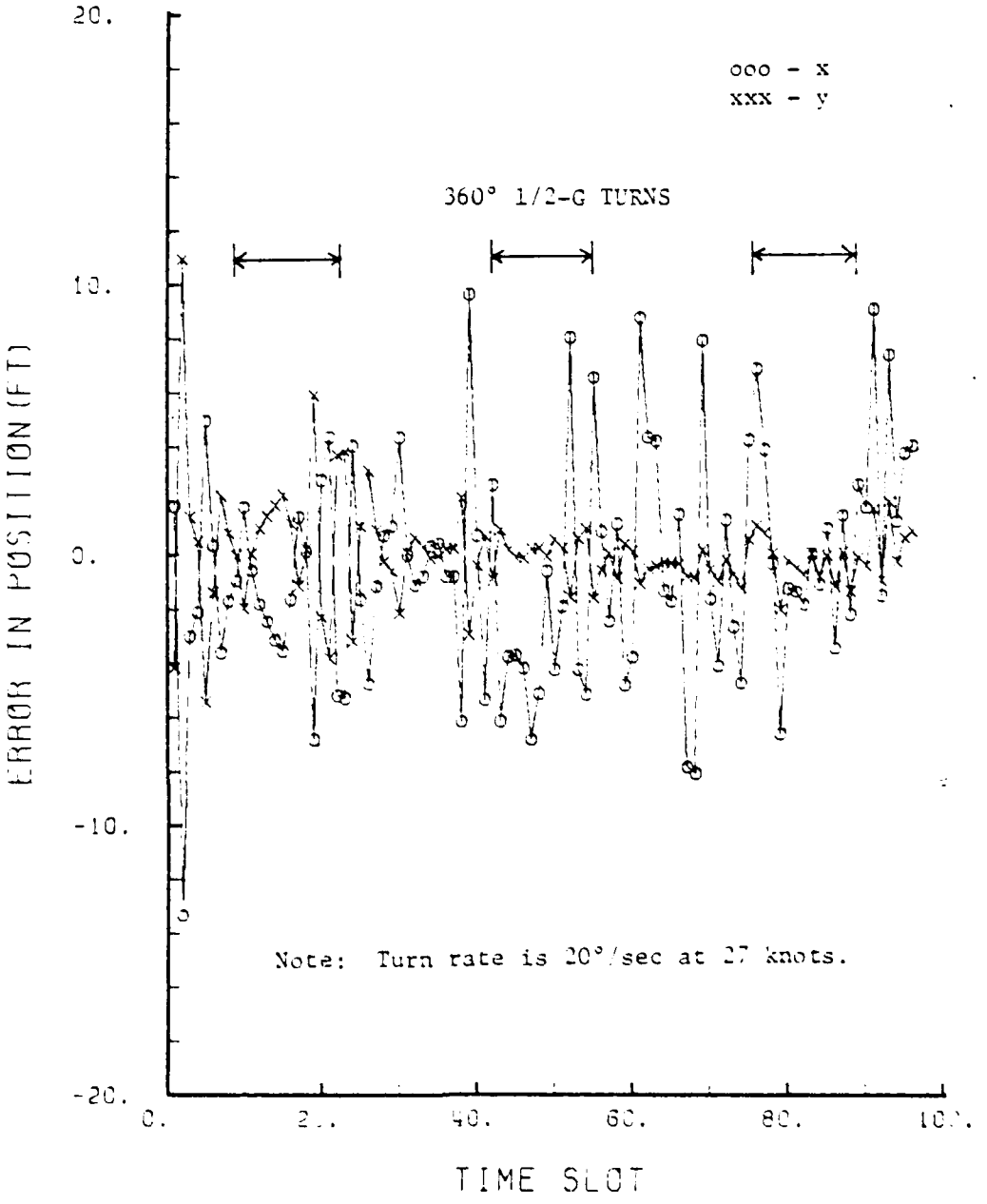


Figure 38. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

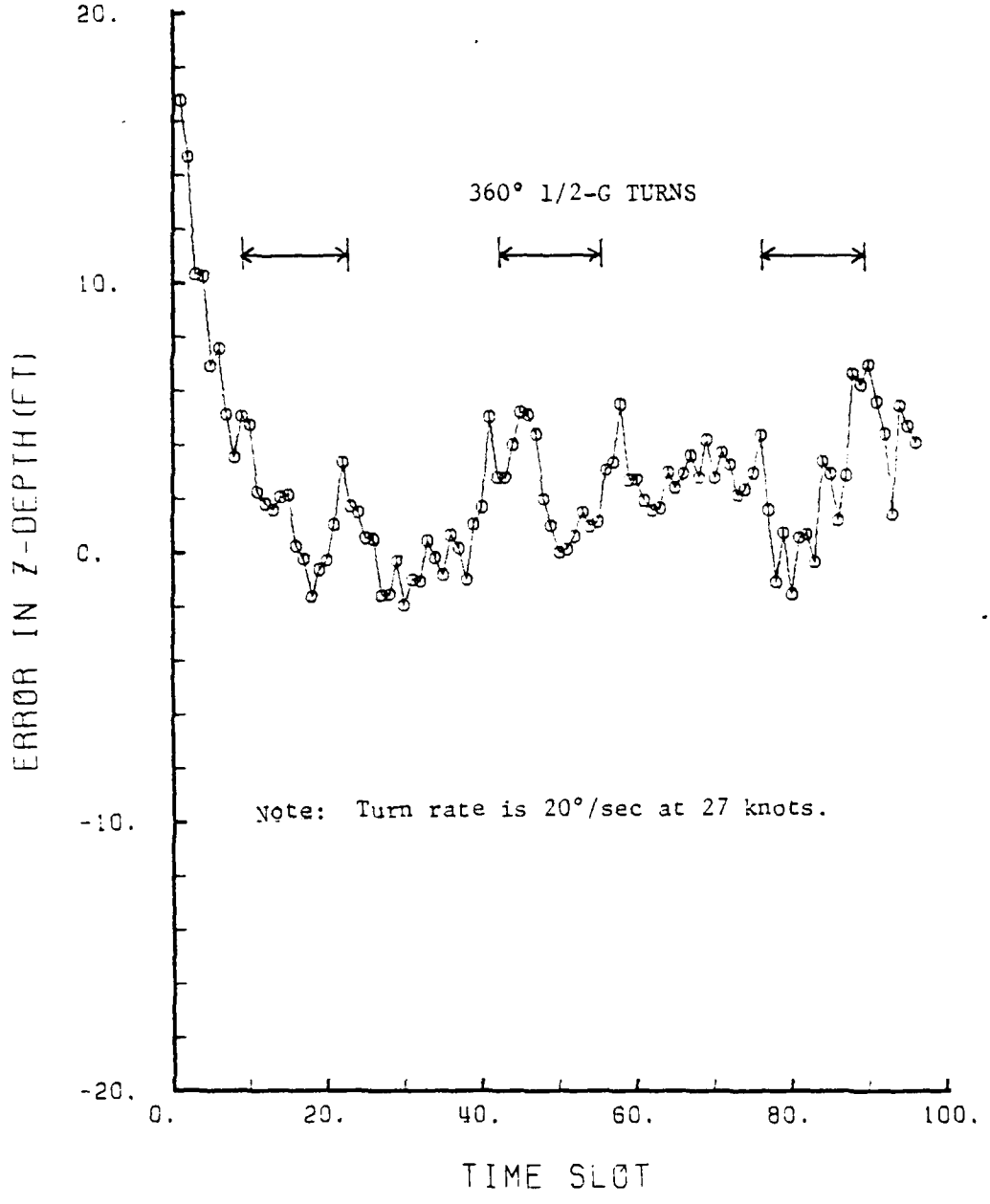


Figure 39. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

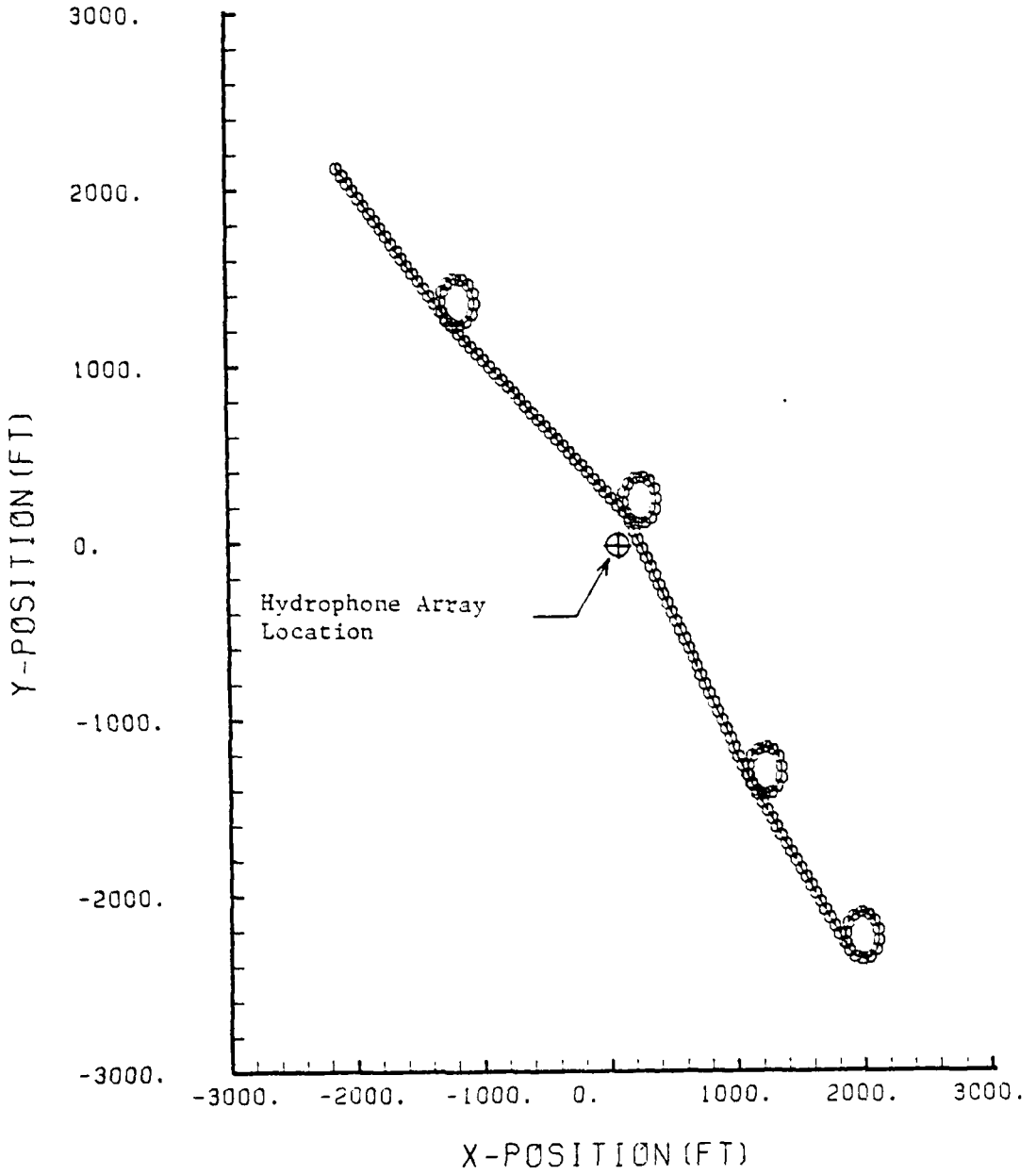


Figure 40. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

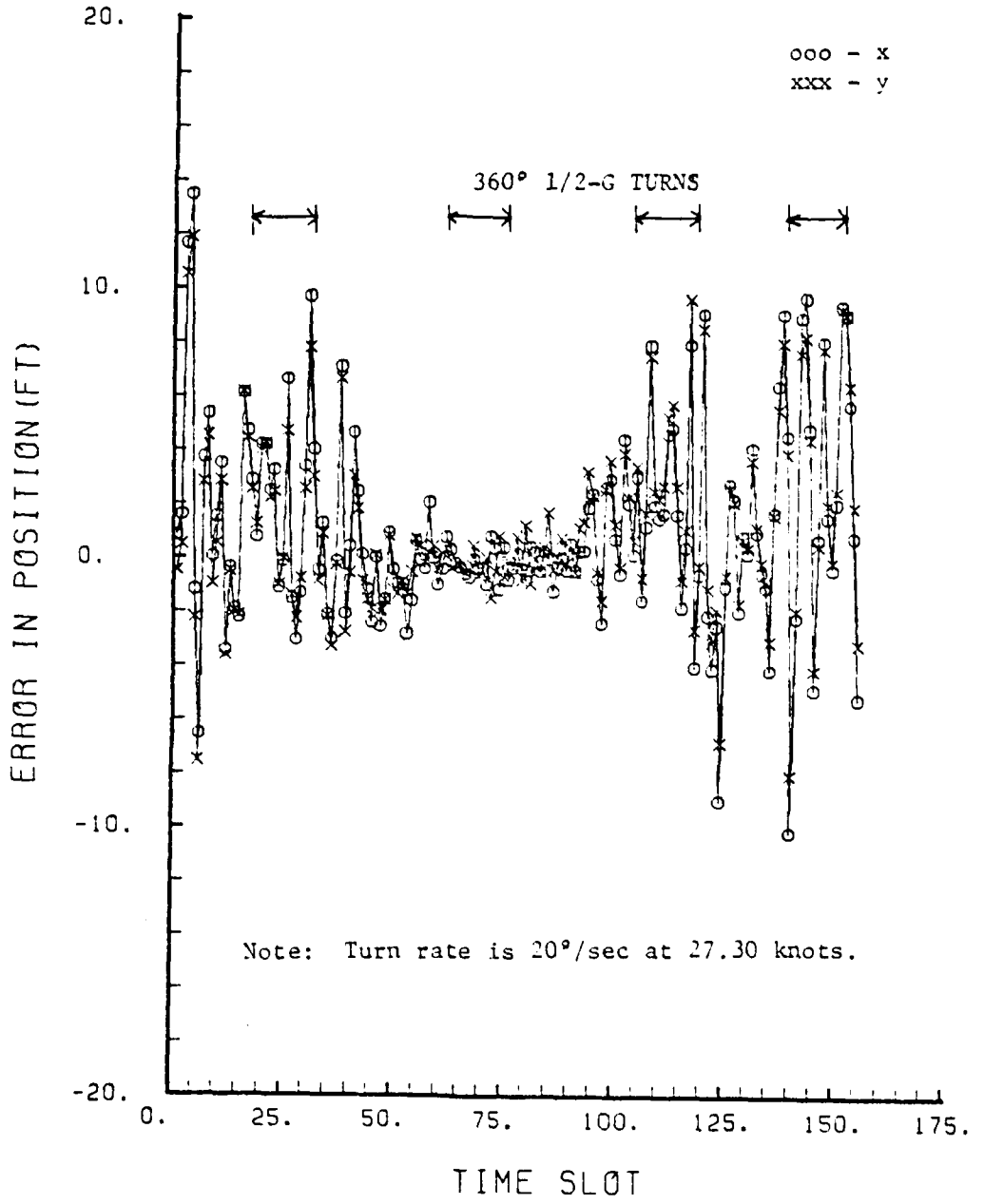


Figure 41. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

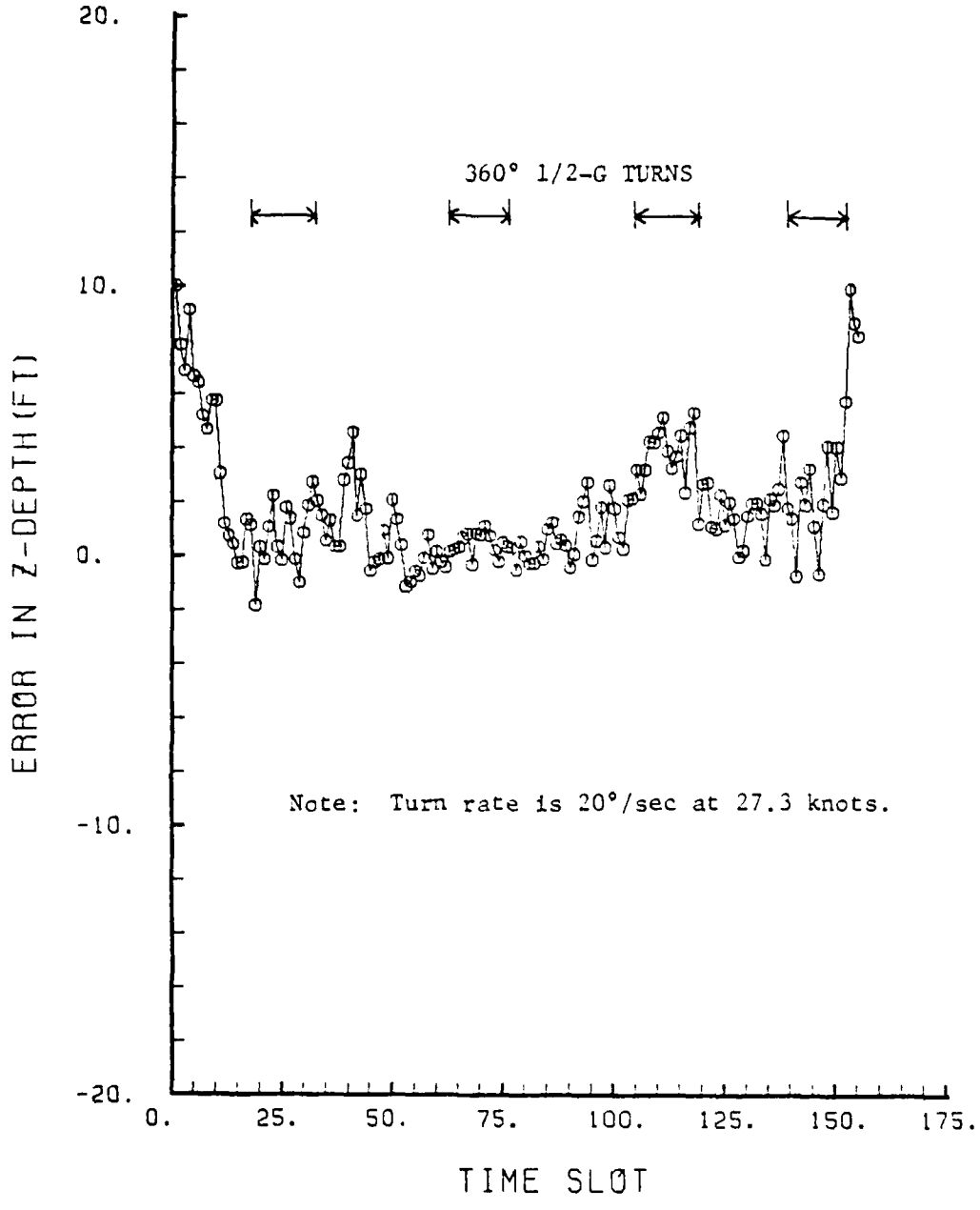


Figure 42. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

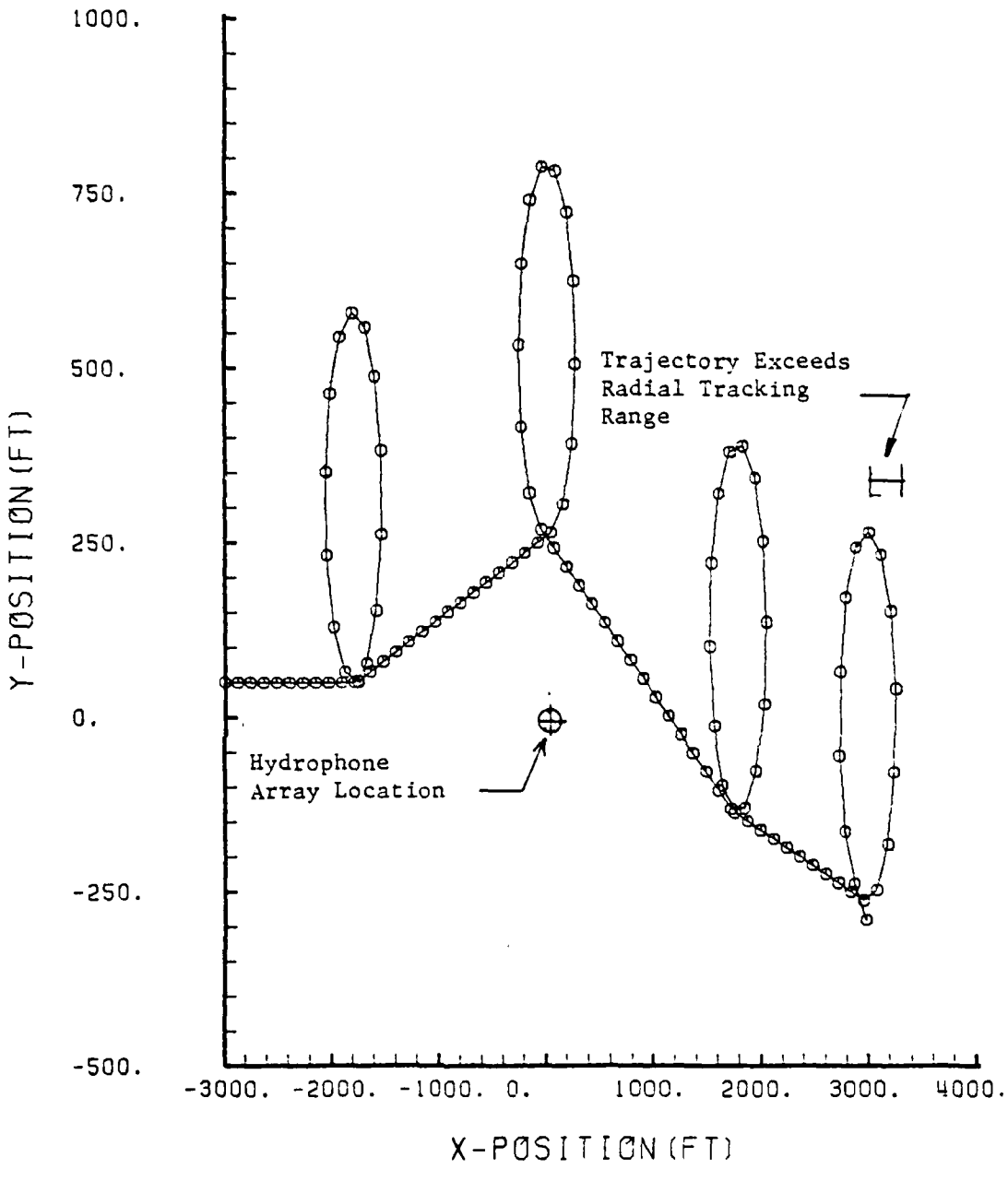


Figure 43. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

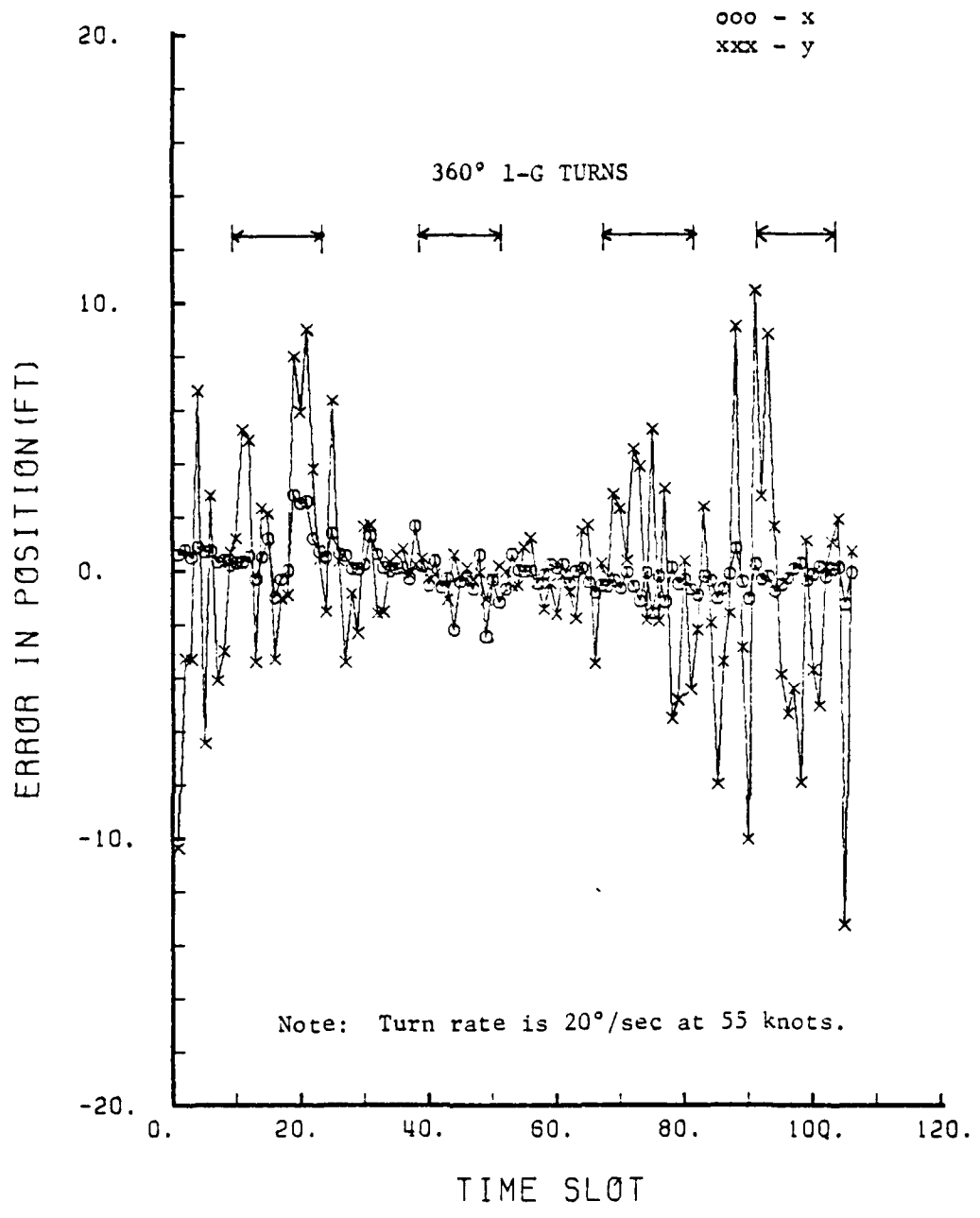


Figure 44. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

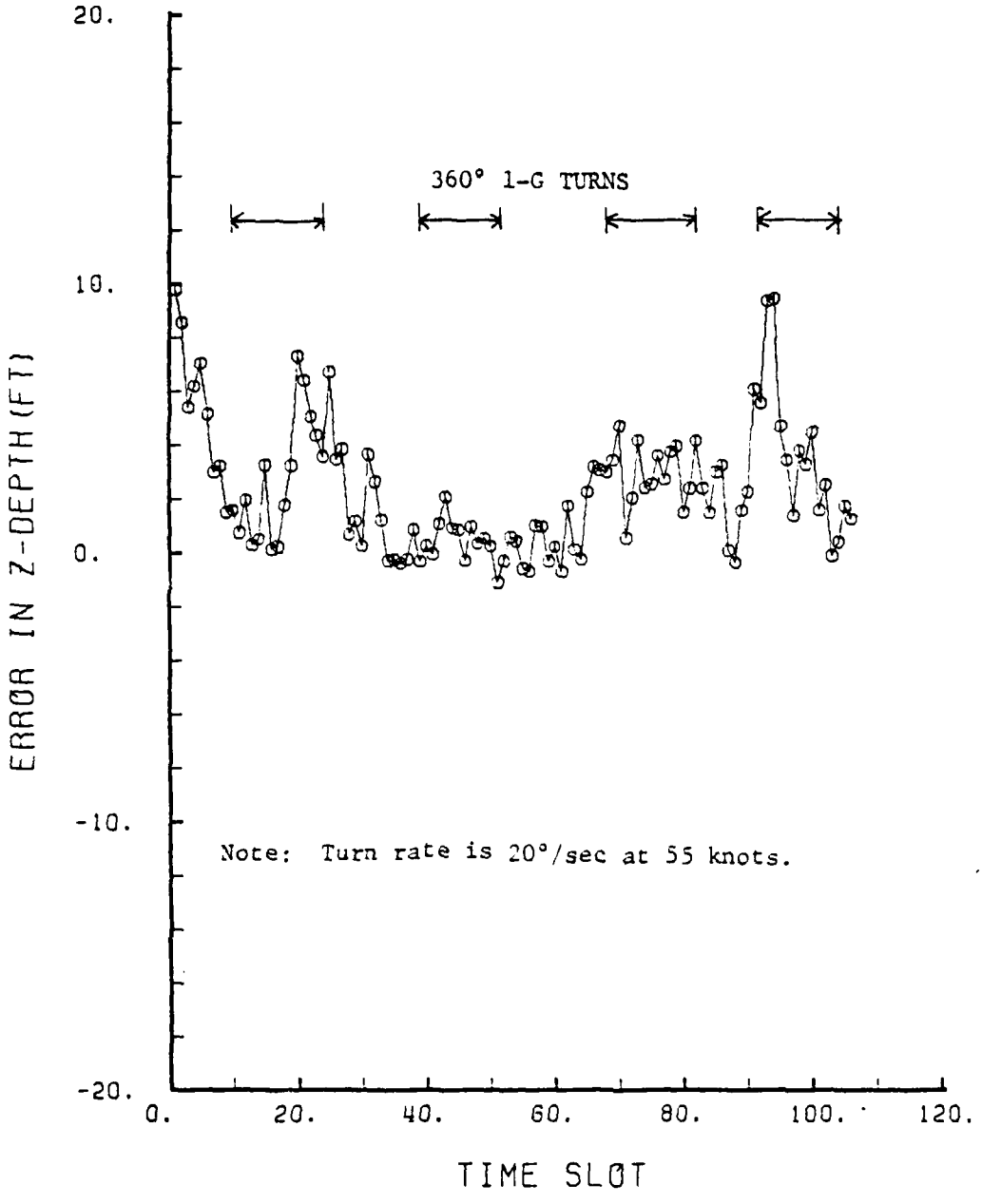


Figure 45. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

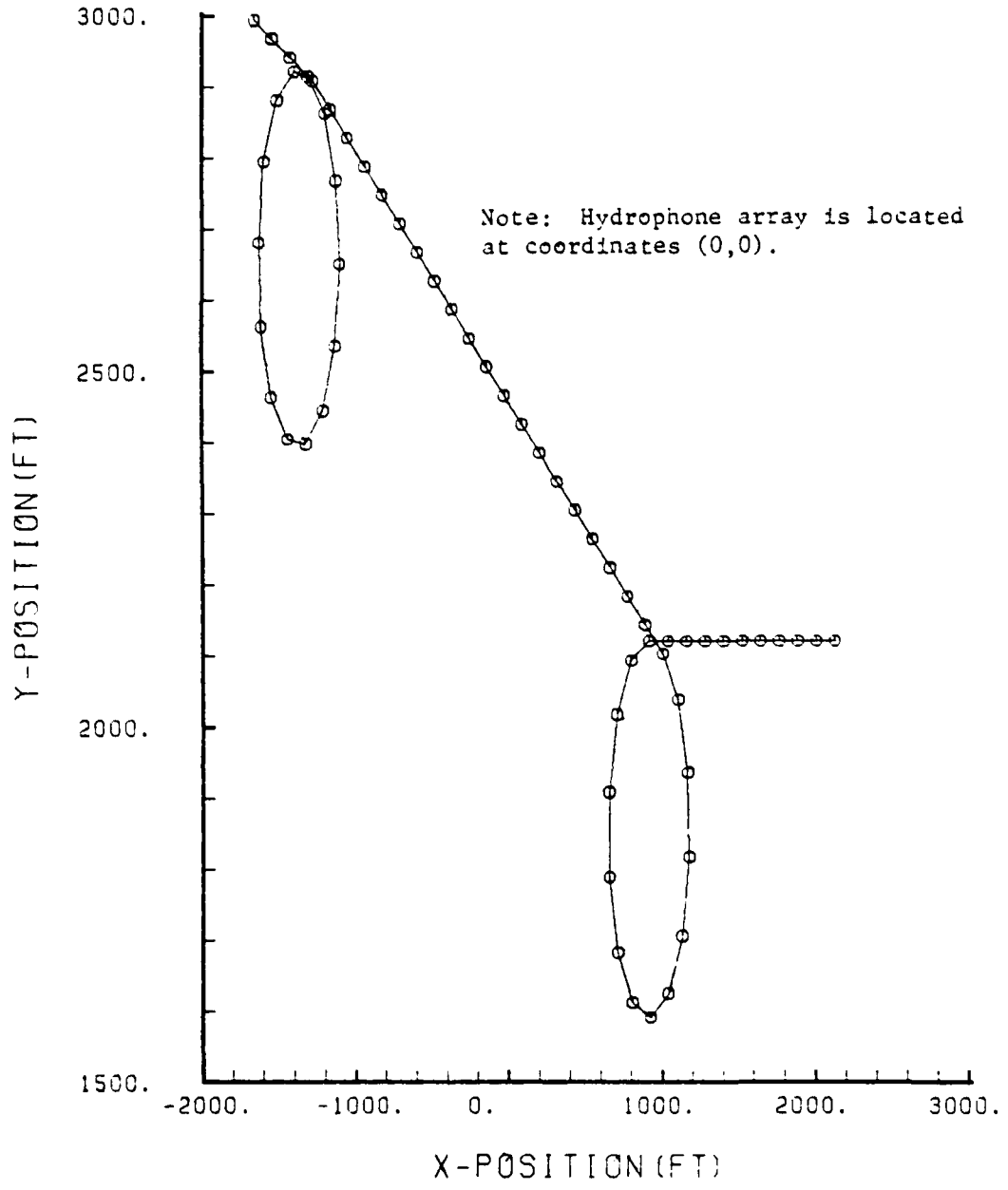


Figure 46. True Trajectory of the Torpedo During a Maneuvering Run in the Area of A Single Array

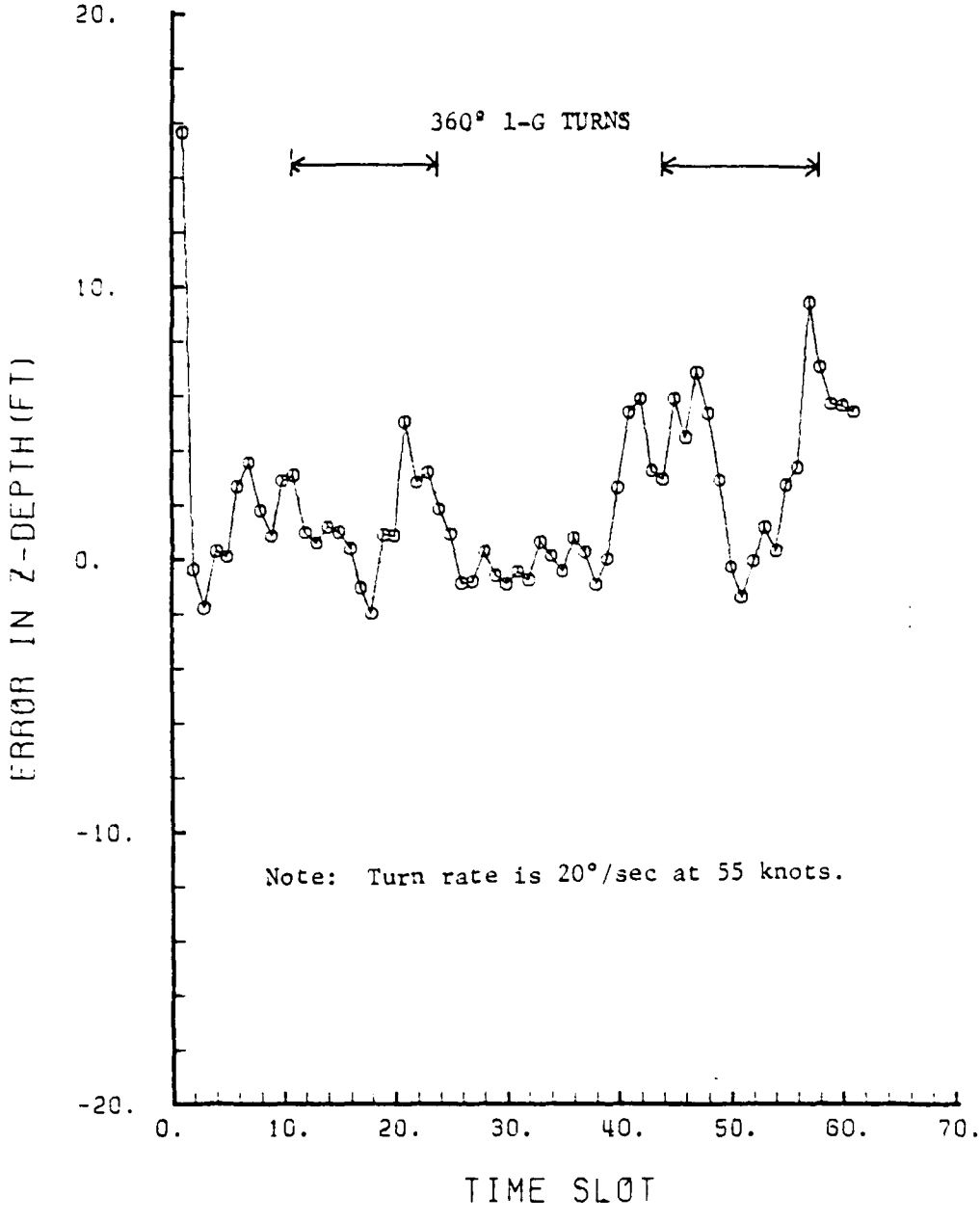


Figure 47. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

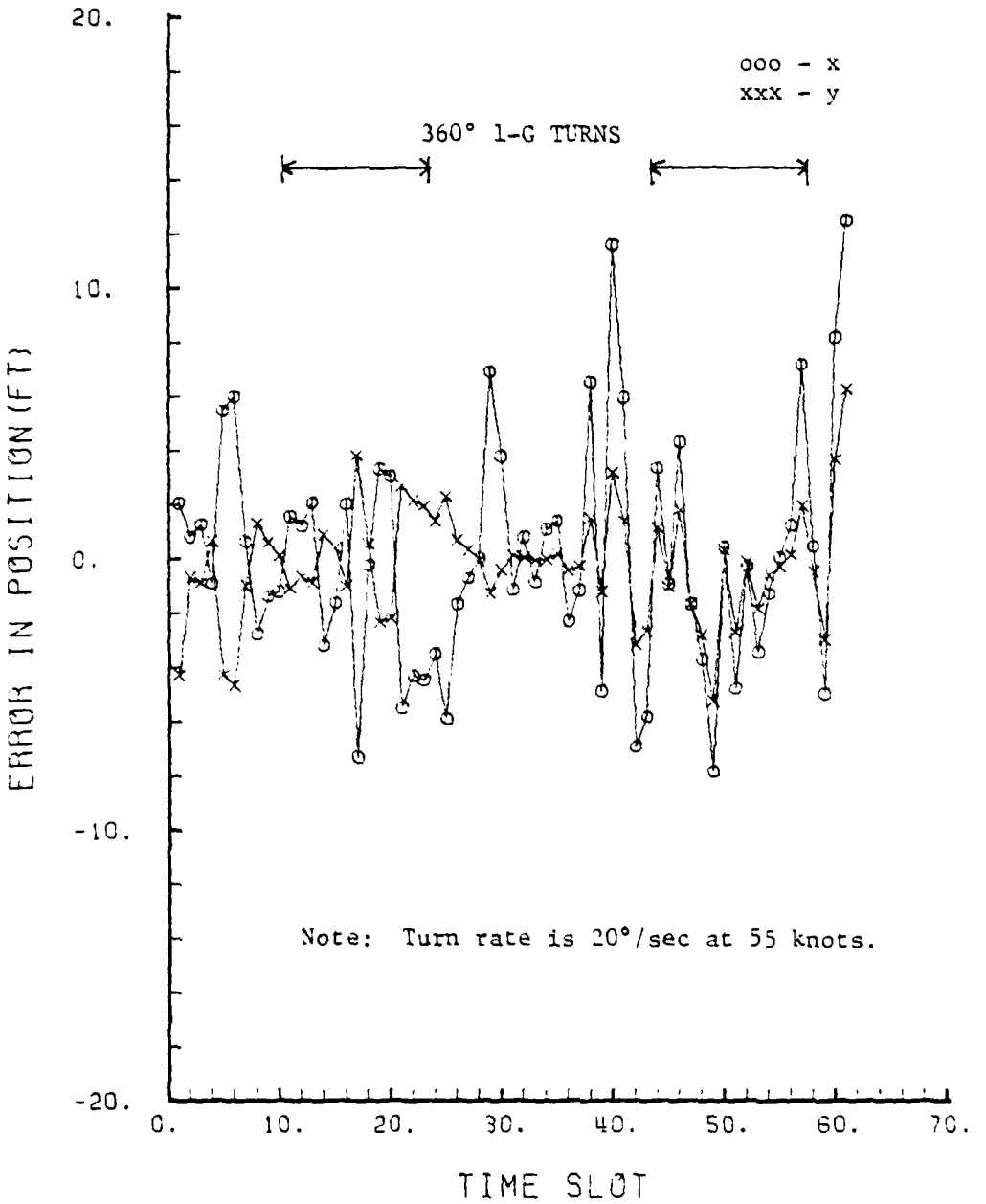


Figure 48. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

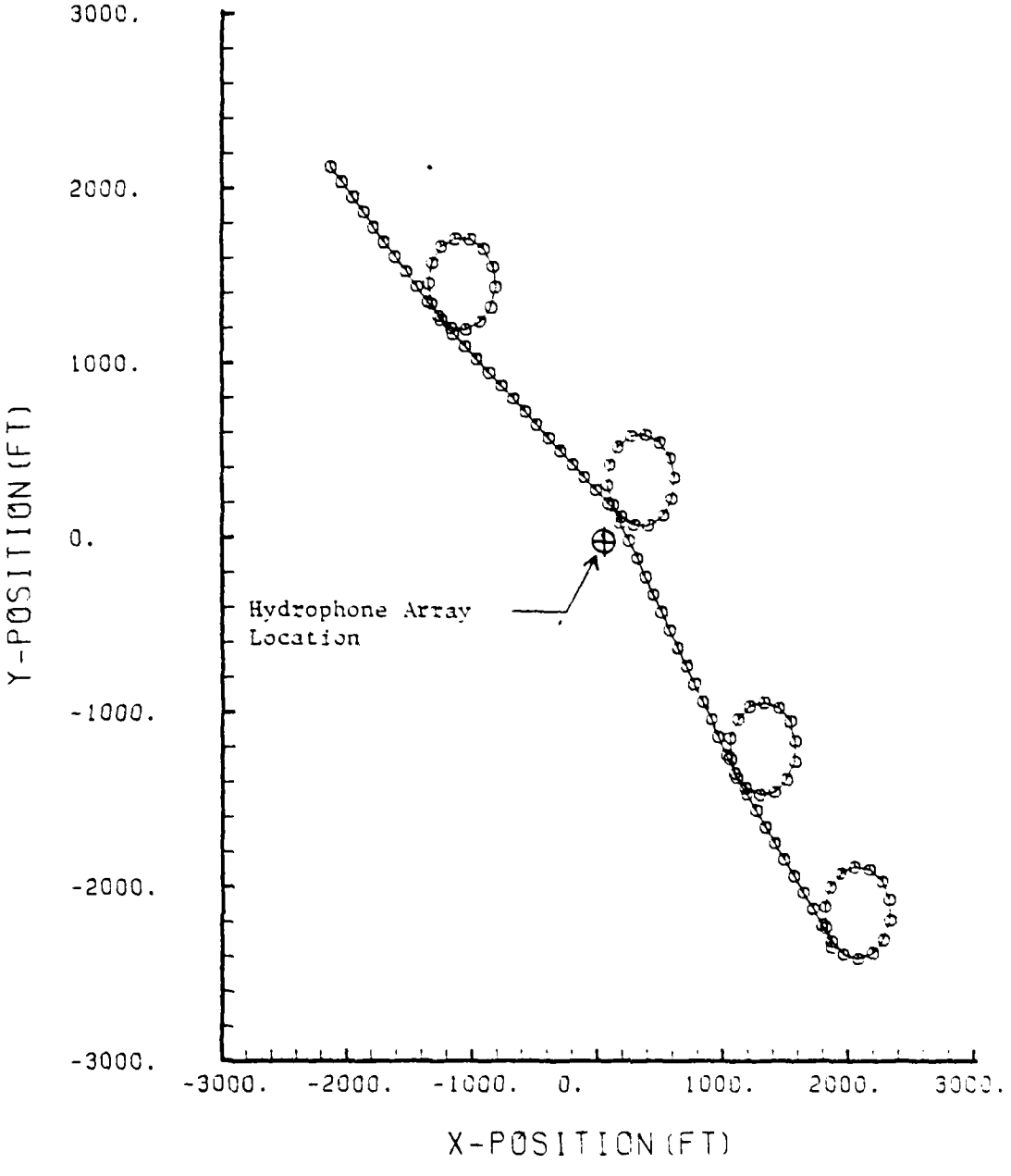


Figure 49. True Trajectory of the Torpedo During a Maneuvering Run in the Area of a Single Array

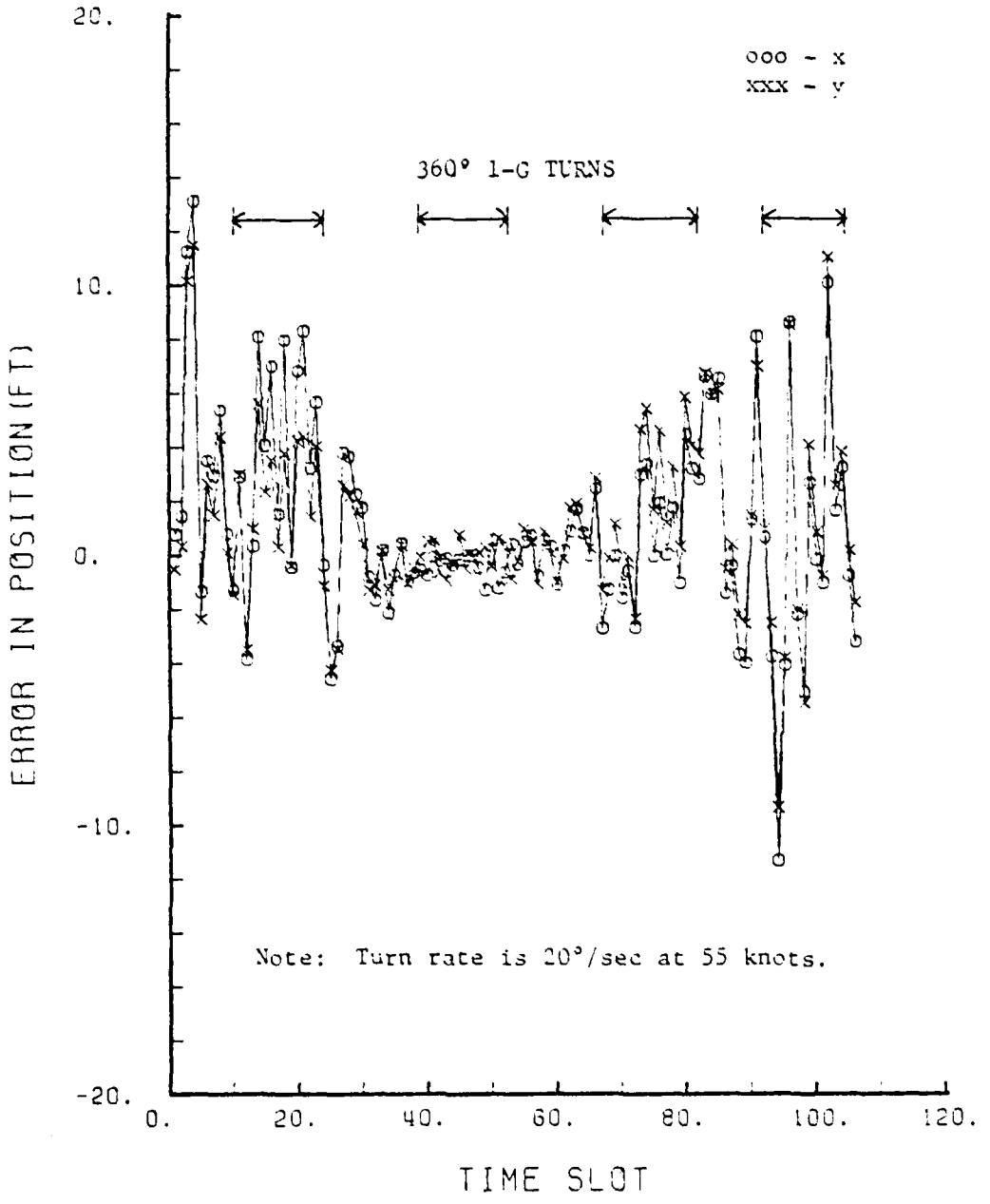


Figure 50. Error in Torpedo Position During a Maneuvering Run in the Area of a Single Array

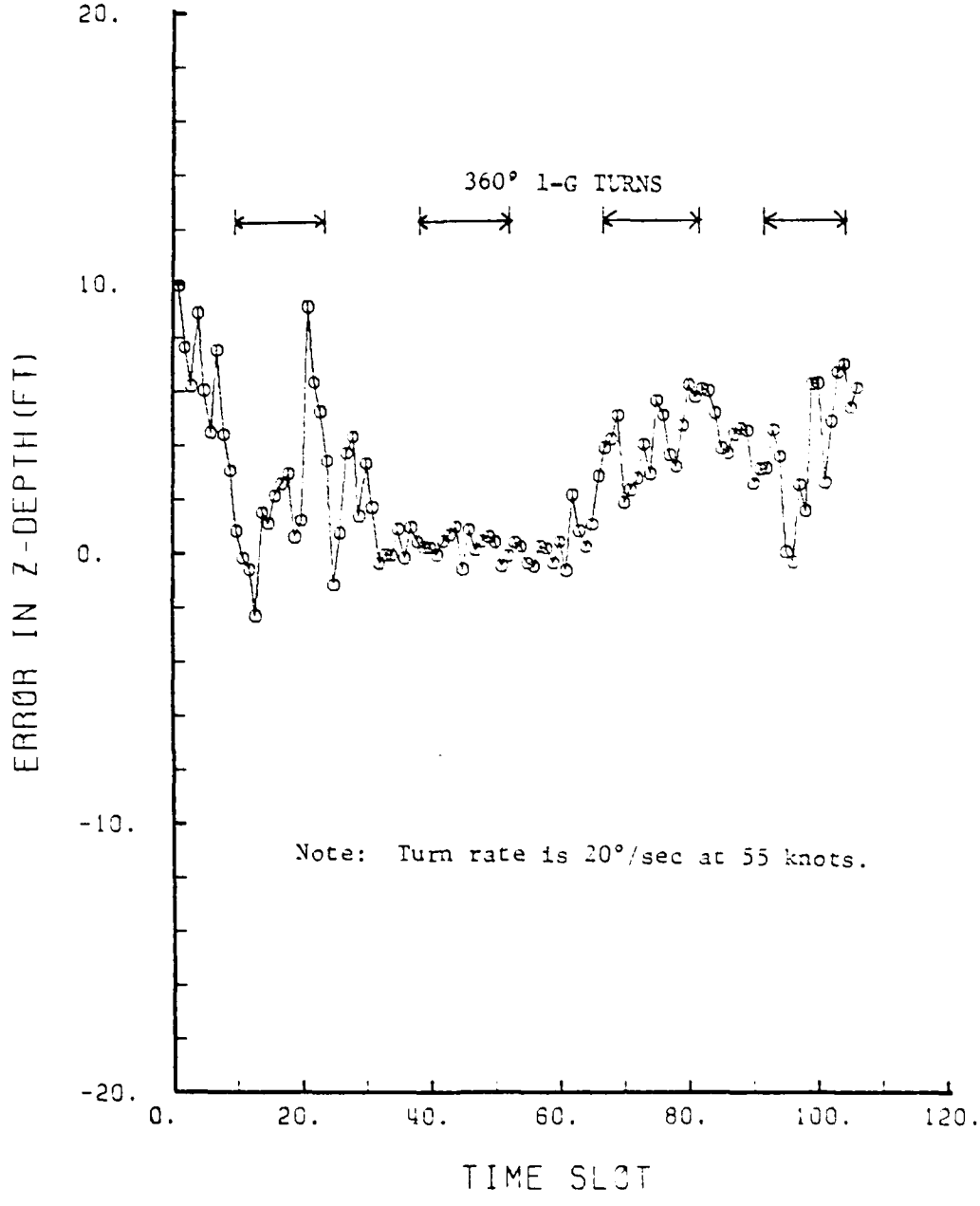


Figure 51. Error in Torpedo Depth During a Maneuvering Run in the Area of a Single Array

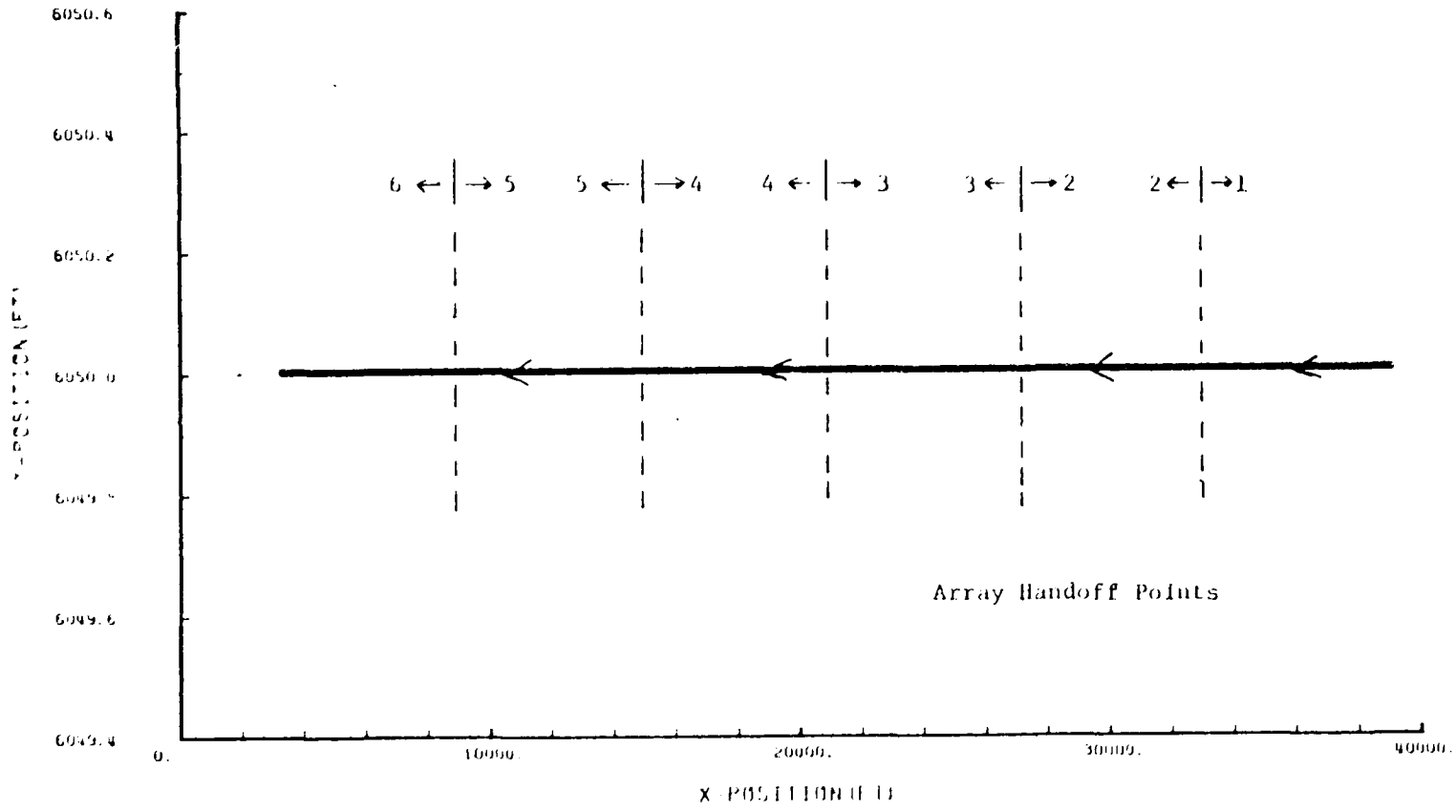


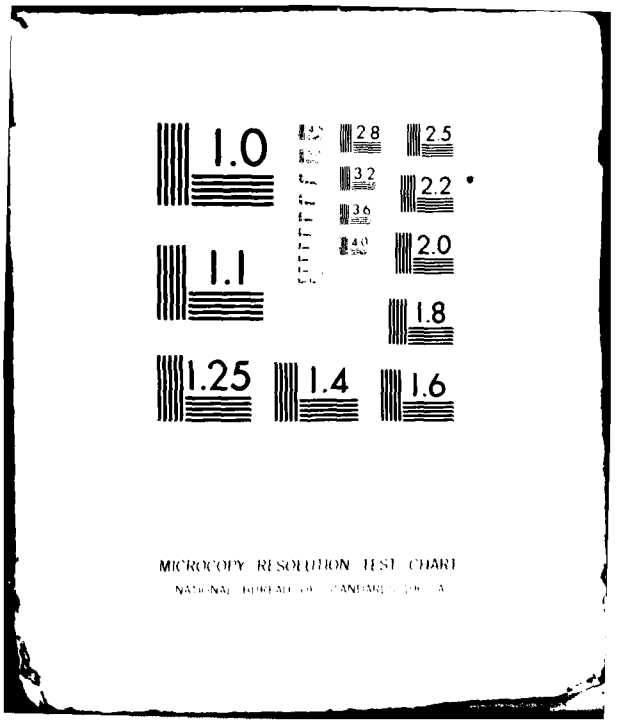
Figure 52. True Trajectory of the Torpedo During a Straight Run Through Multiple Arrays

AD-A092 303 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 17/1
AN APPLICATION OF KALMAN FILTERING TO TORPEDO TRACKING.(U)
SEP 80 P A O'BRIEN

UNCLASSIFIED NL

2 in 2
AD
A092303

END
DATE FILMED
1 81
DTIC


1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART

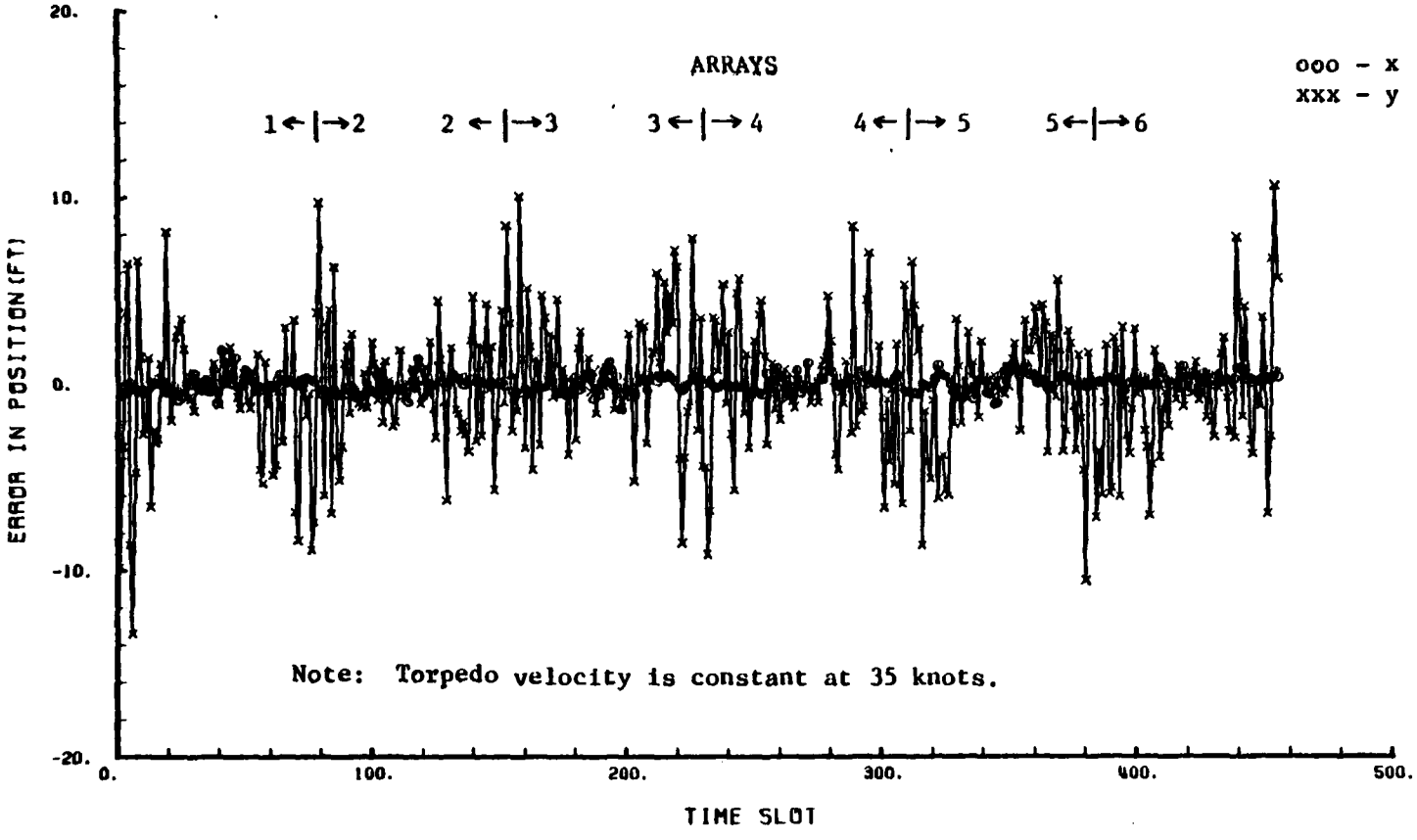


Figure 53. Error in Torpedo Position During a Straight Run Through Multiple Arrays

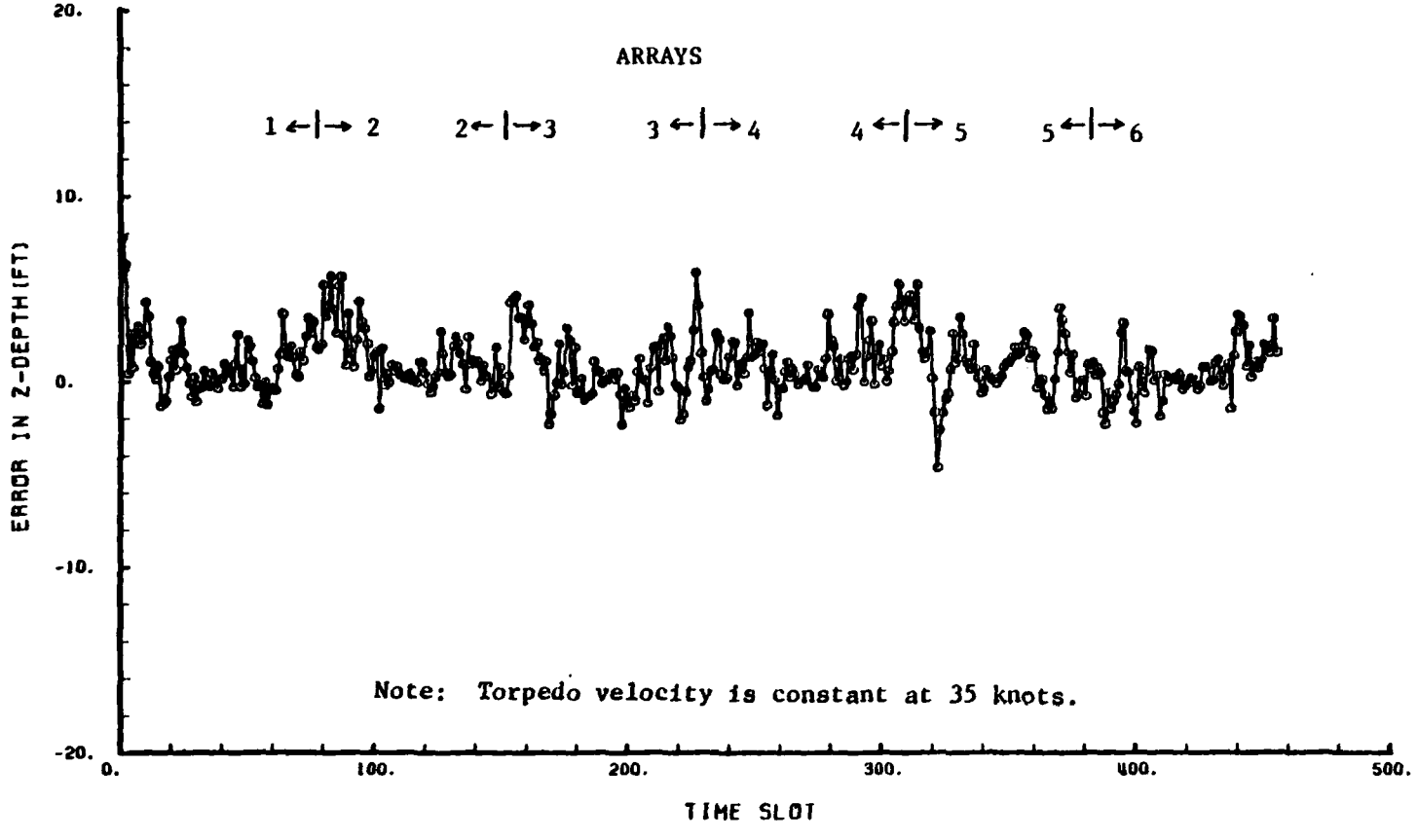


Figure 54. Error in Torpedo Depth During a Straight Run Through Multiple Arrays

APPENDIX A

PROGRAM DESCRIPTION AND FEATURES

The sequential Extended Kalman Filter routine used for torpedo tracking is modularized for ease of implementation. The program is general in nature and many of the parameters of the filter are variable including:

a. The number of states in the filter--N

b. The number of random forcing functions--M

c. The number of measurements--JS

d. Number of time slots--JTIME

The constant matrices PHI, R, COVW, and GAMMA are initialized in the beginning of the program using data statements. The filter is initialized with P(1/0) and X(1/0) (initial covariance of estimation error and states) using subroutine INIT. The first estimate is at time 1 and continues until ITIME = JTIME+1. True measurement times (ZI) are computed using either subroutines TRAJEC or TRAJC3, depending on whether single array or multiple array tracking is implemented. Either subroutine will compute four measurement times (T_C, T_X, T_Y, T_Z) for each time slot. The measurement times are corrupted by zero-mean, white Gaussian noise using the IBM-360 subroutine SNORM. For each of the four time measurements the corresponding row of the linearizing H matrix is calculated using either subroutine CHROW or CHROW3, depending on whether single array

or multiple array tracking is used. The corresponding gain matrix column GI is then found. These row and column values are utilized in forming the covariance of estimation error, PI, for that particular time measurement. Next the estimate of the observation time ZHAT from that particular hydrophone is formed using the subroutine CZHAT or CZHAT3, depending on whether single or multiple array tracking is implemented. The residual ZDIFF(I) = ZIC(I) - ZHAT, is then calcualated. Finally the estimate of the states XI based on one time measurement is calculated and the process is repeated for the next measurement. After the four iterations, four transit time measures are formed using the last estimate of the states, X4, in either subroutine CZHAT or CZHAT3. The corresponding four transit time residuals are then calculated. The average of the absolute value of the time residuals is formed (ZDIFAV) and compared to a preset threshold value in the software. If ZDIFAV is above the threshold, subroutine QFIND is used to calcuate a new Q which is then added to the last covariance of error estimate, P4. The software then reprocesses the transit time measurements for the same time slot. This procedure continues until the ZDIFAV measurement is equal or below the threshold value, at which time the last estimate of the state vector and covariance matrix become the XKK and PKK values for the time slot. Finally, the predictions XKKMI and PKKMI are formed before the process is repeated for the next time slot. PTLOP is used to generate line printer plots and PLOTG is used to generate VERSATEC plots.

A. PROGRAM SUBROUTINES

A brief description of the subroutines that are used with the filter are described below:

1. TRAJEC--This subroutine develops the torpedo trajectory which is used as truth data for the filter. The subroutine generates constant velocity and heading tracks. turns of any desired turn rate and G force can be performed at any time along the torpedo track. The subroutine outputs true transit times, ZI(I), and the X, Y, Z positions, TD(I), of the torpedo for each time slot. The subroutine is used during single array tracking.

2. TRAJC3--This subroutine performs the same function as TRAJEC but is used only during multiple array tracking.

3. INIT--This subroutine generates the initial state vector (X0/-1) and initial covariance matrix (P 0/-1).

4. CHROW--This subroutine computes the appropriate row of the linearizing H matrix. Each row corresponds to one of the four transit time measurements, T_C, T_X, T_Y, T_Z. This subroutine is used during single state array tracking.

5. CHROW3--THis subroutine performs the same functions as CHROW and used during multiple array tracking.

6. CZHAT--This subroutine computes the estimated transit times for the filter. Four time estimates, ZHAT, are calculated corresponding to each of the four true transit times ZI(I). This subroutine is used during single array tracking.

7. CZHAT3--Subroutine performs same functions as CZHAT however it is used only during multiple array tracking.

8. QFIND--This subroutine develops an adative Q matrix which is a function of the torpedo velocity. Three input variables defined in a data statement at the beginning of the program can be adjusted:

aa. SIGACC--Maximum expected horizontal acceleration of the torpedo.

bb. SIGDIV--Maximum expected change in vertical velocity.

cc. SIGCC--Maximum expected turn rate of the torpedo in the horizontal plane.

The values listed in the program were used and kept constant during the simulation tests. If the user desires not to use the adaptive Q subroutine software code is provided at the beginning of the program to calculate a constant Q matrix.

9. SNORM--This is an IBM-360 subroutine contained in the IMSL library. The routine generates zero mean white Gaussian noise with an RMS value normalized to 1. The main program scales the noise and adds it to the transit times measurements.

10. PLOTP--This is an IBM-360 subroutine used to generate the line printer plots. Information on this subroutine can be obtained from the IMSL library.

11. PLOTG--This is an IBM-360 subroutine used to generate the VERSATEC plots. Information on this subroutine can be obtained from the IMSL library.

B. UTILITY PROGRAMS

These subroutines were designed to be used for repetitive matrix and vector manipulations:

1. PROD--multiplying two matrices
2. MMULT--multiplying a matrix and a vector
3. VMULT--multiplying two vectors
4. TRANS--transposing a matrix
5. ADD--adding two matrices

```
      DOUBLE PRECISION XKKM1,PKKM1,PKK,XKK,PHI,GAMMA,R,PHIPKK,PHIT,      THE00000
     1PKTEMP,HROW,PI,PDUM,QTEMP,GNUM,GI,XI,ZI,GAMMAT,Q,COVW,ZC,ZIC,      THE00005
     2ZCS,ZHAT,DATA,ZDIFF,GDENOM,GDTEMP,ZDIFAV                           THE00010
      COMMON XKKM1(5),PKKM1(5,5),PKK(5,5),XKK(5)                         THE00015
      DIMENSION PHI(5,5),GAMMA(5,3),R(4,4),PHIPKK(5,5),PHIT(5,5),        THE00020
     1PKTEMP(5,5),HROW(5),PI(5,5),PDUM(5,5),QTEMP(5,3),X1(510),          THE00025
     2GNUM(5),GI(5),XI(5),ZI(4),GAMMAT(3,5),Q(5,5),COVW(3,3),            THE00030
     3X5(510),KOUNT(510),AKOUNT(510),PVEC(510),PI(510,5,5),X3(510),      THE00035
     4TRUX(510),TRUY(510),TRUZ(510),TD(3),C1(2),C2(2),C3(2),             THE00040
     5C4(2),C5(2),C6(2),C7(2),XP(5,510),XDIFF1(510),XDIFF3(510),         THE00045
     6XDIFF5(510),ZC(4),HYDRO(6,12),XB(4),YB(4),ZB(4),DATA(23),ZDIFF(4),THE00050
     7GX(510,4),GY(510,4),ZDIFF1(510,4),ZDM(2),GXM(2),GYM(2),ZIC(4)      THE00055
C                                                                        THE00060
CCC   INITIALIZE CONSTANTS,HYDROPHONE MATRIX,PHI,R,COVW,GAMMA,           THE00065
CCC   AND DATA FOR TRAJECTORY                                            THE00070
C                                                                        THE00075
      DATA N/5/,M/3/,JS/4/,I8/1/,NZDIFF/4/,JTIME/454/,                   THE00080
     1  SIGACC/36.2/,SIGDIV/1.0/,SIGCC/22.2/,ZC/4*0.0/,                  THE00085
     2 HYDRO/36000.,30000.,24000.,18000.,12000.,6000.,6*6000.,          THE00090
     3   6*0.0,36030.,30030.,24030.,18030.,12030.,6030.,6*6000.,        THE00095
     4   6*0.0,36000.,30000.,24000.,18000.,12000.,6000.,6*6030.,        THE00100
     5   6*0.0,36000.,30000.,24000.,18000.,12000.,7*6000.,6*30./        THE00105
      DATA PHI/1.0,4*0.0,1.31,1.0,5*0.0,1.0,4*0.0,1.31,1.0,5*0.0,1.0/,   THE00110
     1 R/1.0E-10,4*0.0,1.0E-10,4*0.0,1.0E-10,4*0.0,1.0E-10/,            THE00115
     2 COVW/1036.84,3*0.0,1036.84,3*0.0,1.0/,                           THE00120
     3 GAMMA/.86,1.31,5*0.0,.86,1.31,5*0.0,1.31/,                       THE00125
C                                                                        THE00130
CCC   DATA STATEMENT CONTAINS VALUES FOR TRAJEC OR TRAJC3                THE00135
CCC   DATA STATEMENT DEFINED IN SUBROUTINE TRAJEC OR TRAJC3              THE00140
C                                                                        THE00145
     4DATA/39000.,6050.,300.,-60.,0.0,3*0.0,                            THE00150
     13*0.0,4.71,.174,.1,8.1,600.,800./                                 THE00155
      SIGCC=SIGCC*3.14159/180.                                           THE00160
C                                                                        THE00165
CCC   THE NEXT TWO STATEMENTS ARE REQUIRED FOR                           THE00170
CCC   SUBROUTINE SNORM WHICH GENERATES WHITE NOISE                       THE00175
C                                                                        THE00180
      CALL OVFLOW                                                        THE00185
      I9=6735476                                                         THE00190
C                                                                        THE00195
CCC   LOAD X(0/-1),P(0/-1)                                               THE00200
C                                                                        THE00205
      CALL INIT(XKKM1,PKKM1)                                             THE00210
C                                                                        THE00215
CC    GET TRANSPOSES                                                     THE00220
      CALL TRANS(GAMMA,N,M,GAMMAT)                                       THE00225
      CALL TRANS(PHI,N,N,PHIT)                                           THE00230
      WRITE(6,163)                                                       THE00235
```

```
 163   FORMAT('0','R MATRIX')                                          THE00240
      DO 264 I=1,4                                                     THE00245
 264  WRITE(6,136)(R(I,J),J=1,4)                                       THE00250
 136   FORMAT(4F14.11)                                                 THE00255
C     WRITE(6,265)                                                     THE00260
C265  FORMAT('0','COV W MATRIX')                                       THE00265
C     DO 266 I=1,3                                                     THE00270
C266  WRITE(6,267)(COVW(I,J),J=1,3)                                    THE00275
C267  FORMAT(3F11.6)                                                   THE00280
C                                                                      THE00285
C CALCULATE THE Q MATRIX                                               THE00290
C                                                                      THE00295
C     CALL PROD(GAMMA,COVW,N,M,M,QTEMP)                                THE00300
C     CALL PROD(QTEMP,GAMMAT,N,M,N,Q)                                  THE00305
C     WRITE(6,100)                                                     THE00310
 100  FORMAT('0','Q MATRIX')                                           THE00315
C     DO 101 LI=1,N                                                    THE00320
C101  WRITE(6,102)(Q(LI,LJ),LJ=1,N)                                    THE00325
C102  FORMAT(1X,5F15.4)                                                THE00330
C                                                                      THE00335
C START THE TIME SLOT LOOP AND SET ARRAY HANDOFF POINT                 THE00340
C                                                                      THE00345
      ITIME=JTIME+1                                                    THE00350
      XT=36000.                                                        THE00355
      SW=XT-3000.                                                      THE00360
        DO 99 KK=1,ITIME                                               THE00365
C            WRITE(6,6) KK                                             THE00370
 6           FORMAT (//1X,100('*'),'TIME=',I4//)                       THE00375
C                                                                      THE00380
C     GET HYDROPHONE ARRAY COORDINATES                                 THE00385
C                                                                      THE00390
      DO 600 I3=1,4                                                    THE00395
      XB(I3)=HYDRO(I8,3*I3-2)                                          THE00400
      YB(I3)=HYDRO(I8,3*I3-1)                                          THE00405
 600  ZB(I3)=HYDRO(I8,3*I3)                                            THE00410
      IF(XKKM1(1).GT.SW)GO TO 610                                      THE00415
      I8=I8+1                                                          THE00420
      WRITE(6,759)I8,KK                                                THE00425
 759  FORMAT ('0',15('*'),'ARRAY',1X,I2,'STARTS TRACKING AT TIME',1X,I3,THE00430
     1 15('*')/)                                                       THE00435
      SW=SW-6000.                                                      THE00440
      XT=XT-6000.                                                      THE00445
C     WRITE(6,67)                                                      THE00450
C67   FORMAT('0','XKKM1')                                             THE00455
C     DO 51 LA=1,N                                                     THE00460
C51   WRITE(6,50) XKKM1(LA)                                           THE00465
C50   FORMAT(1X,5F14.4)                                               THE00470
C                                                                      THE00475
```

```
C COMPUTE THE TRUE TIMES AND TRUE POSITIONS                              THE00480
C                                                                        THE00485
 610  CALL TRAJC3(KK,DATA,ZI,TD,XB,YB,ZB)                               THE00490
C610  CALL TRAJEC(KK,DATA,ZI,TD)                                        THE00495
      A14=PHI(1,2)                                                       THE00500
        TRUX(KK)=TD(1)                                                   THE00505
        TRUY(KK)=TD(2)                                                   THE00510
        TRUZ(KK)=TD(3)                                                   THE00515
C                                                                        THE00520
C       FIRST GET HROW-CALCULATE GAIN,ESTIMATE,COVARIANCE OF             THE00525
C       ERROR BASED ON ONE TIME MEASUREMENT-TC,TX,TY, TZ/                THE00530
C                                                                        THE00535
 711       DO 97 I=1,JS                                                  THE00540
         CALL CHROW3(I,HROW,XB,YB,ZB)                                    THE00545
C        CALL CHROW(I,HROW)                                              THE00550
          CALL MMULT(PKKM1,HROW,N,N,GNUM)                                THE00555
          CALL VMULT(HROW,GNUM,N,GDTEMP)                                 THE00560
          GDENOM=GDTEMP+R(I,I)                                           THE00565
             DO 16 IX=1,N                                                THE00570
 16          GI(IX)=GNUM(IX)/GDENOM                                      THE00575
C                                                                        THE00580
C         THIS IS THE FIRST GAIN COLUMN                                  THE00585
C         CALCULATE THE COVARIANCE OF ERROR PI                           THE00590
C                                                                        THE00595
            DO 77 IP=1,N                                                 THE00600
               DO 79 JP=1,N                                              THE00605
               PDUM(IP,JP)=(-1.*GI(IP))*HROW(JP)                         THE00610
             IF(IP.EQ.JP)PDUM(IP,JP)=1.+PDUM(IP,JP)                      THE00615
 79           CONTINUE                                                   THE00620
 77          CONTINUE                                                   THE00625
          CALL PROD (PDUM,PKKM1,N,N,N,PI)                                THE00630
C                                                                        THE00635
C         CALCULATE FIRST MEASUREMENT PREDICTION                         THE00640
C                                                                        THE00645
C         CALL CZHAT(I,ZHAT)                                             THE00650
          CALL CZHAT3(I,ZHAT,XB,YB,ZB)                                   THE00655
C                                                                        THE00660
C         GET WHITE NOISE,SCALE AND ADD TO TRUE MEASUREMENT TIME         THE00665
C                                                                        THE00670
          CALL SNORM(I9,ZC,4)                                            THE00675
          ZCS=ZC(1)*.00001                                               THE00680
          ZIC(I)=ZCS+ZI(I)                                               THE00685
          ZDIFF(I)=ZIC(I)-ZHAT                                           THE00690
C                                                                        THE00695
C     THESE STATEMENTS ARE FOR ADDITIONAL PLOTTING                       THE00700
C                                                                        THE00705
      GX(KK,I)=GI(1)                                                     THE00710
      GY(KK,I)=GI(3)                                                     THE00715
```

```
      ZDIFF1(KK,I)=ZDIFF(I)                                              THE00720
      IF ((K<.GT.1).OR.(I.GT.1)) GO TO 73                                THE00725
         ZDM(1)=ZDIFF(I)                                                 THE00730
         ZDM(2)=ZDIFF(I)                                                 THE00735
         GXM(1)=GI(1)                                                    THE00740
         GXM(2)=GI(I)                                                    THE00745
         GYM(1)=GI(3)                                                    THE00750
         GYM(2)=GI(3)                                                    THE00755
         GO TO 74                                                        THE00760
  73  CONTINUE                                                           THE00765
      IF (ZDIFF(I).GT.ZDM(1)) ZDM(1)=ZDIFF(I)                            THE00770
      IF (ZDIFF(I).LT.ZDM(2)) ZDM(2)=ZDIFF(I)                            THE00775
      IF (GI(1).GT.GXM(1)) GXM(1)=GI(I)                                  THE00780
      IF (GI(1).LT.GXM(2)) GXM(2)=GI(1)                                  THE00785
      IF(GI(3).GT.GYM(1)) GYM(1)=GI(3)                                   THE00790
      IF (GI(3).LT.GYM(2)) GYM(2)=GI(3)                                  THE00795
  74  CONTINUE                                                           THE00800
C                                                                        THE00805
C     COMPUTE THE GATE FOR ERRONEOUS TIME MEASUREMENTS                   THE00810
C                                                                        THE00815
      PK1=DABS(PI(1,1))                                                  THE00820
      PK3=DABS(PI(3,3))                                                  THE00825
      PK5=DABS(PI(5,5))                                                  THE00830
      IF((PK1.GE.PK3).AND.(PK1.GE.PK5))P=PK1                             THE00835
      IF((PK3.GE.PK1).AND.(PK3.GE.PK5))P=PK3                             THE00840
      IF((PK5.GE.PK1).AND.(PK5.GE.PK3))P=PK5                             THE00845
      PGATE=SQRT(P)/4860.                                                THE00850
      RGATE=DSQRT(DABS(R(I,I)))                                          THE00855
      GATE=3.*(PGATE+RGATE)                                              THE00860
C                                                                        THE00865
C     EDIT INVALID TIME MEASUREMENTS                                     THE00870
C                                                                        THE00875
      IF (ZDIFF(I).LT.GATE) GO TO 500                                    THE00880
      WRITE(6,501)KK                                                     THE00885
 501  FORMAT('0','GATE HAS BEEN EXCEEDED TIME',I4)                       THE00890
      DO 502 LG=1,N                                                      THE00895
 502  GI(LG)=0.0                                                         THE00900
C                                                                        THE00905
C      TAG INVALID TIME MEASUREMENT                                      THE00910
C                                                                        THE00915
      ZDIFF(I)=999.                                                      THE00920
C                                                                        THE00925
C     CALCULATE THE ESTIMATE BASED ON ONE MEASUREMENT                    THE00930
C                                                                        THE00935
 500          DO 17 IZ=1,N                                               THE00940
 17          XI(IZ)=XKKMI(IZ)+GI(IZ)*ZDIFF(I)                           THE00945
          IF(I.EQ.4)GO TO 56                                             THE00950
   12       DO 19  IQ=1,N                                               THE00955
```

```
 19          XKKM1(IQ)=XI(IQ)                                              THE00960
              DO 23 IQ=1,N                                                 THE00965
                DO 18 JQ=1,N                                               THE00970
 18               PKKM1(IQ,JQ)=PI(IQ,JQ)                                   THE00975
 23           CONTINUE                                                     THE00980
C     WRITE(6,31)                                                         THE00985
C31   FORMAT('0')                                                         THE00990
 97         CONTINUE                                                      THE00995
C                                                                         THE01000
C       NOTE-CALLED ORIGINAL X(0/-1),XKKM1. UPDATED AFTER 1               THE01005
C       MEASUREMENT CALLED IT XI, THEN MADE XKKM1=XI AND WENT             THE01010
C       THRU ITERATION AGAIN. AFTER YOU HAVE UPDATED XKKM1 FOR            THE01015
C       EACH MEASUREMENT XKK=XI AND PKK=PI                                THE01020
C                                                                         THE01025
 56       DO 57 ID=1,N                                                    THE01030
          XKK(ID)=XI(ID)                                                  THE01035
           XKKM1(ID)=XI(ID)                                               THE01040
            DO 58 JD=1,N                                                  THE01045
           PKK(ID,JD)=PI(ID,JD)                                           THE01050
 58        PKKM1(ID,JD)=PI(ID,JD)                                         THE01055
 57        CONTINUE                                                       THE01060
C                                                                         THE01065
C     RECALCULATE TIME MEASUREMENTS,FORM ABSOLUTE VALUE OF RESIDUALS      THE01070
C                                                                         THE01075
      DO 81 I=1,4                                                         THE01080
C                                                                         THE01085
C     EDIT INVALID TIME MEASUREMENTS FOR ADAPTIVE MANEUVER ROUTINE        THE01090
C                                                                         THE01095
      IF (ZDIFF(I).GE.999.) GO TO 82                                      THE01100
C     CALL CZHAT(I,ZHAT)                                                  THE01105
      CALL CZHAT3(I,ZHAT,XB,YB,ZB)                                        THE01110
      ZDIFF(I)=DABS(ZIC(I)-ZHAT)                                          THE01115
      GO TO 81                                                            THE01120
 82   ZDIFF(I)=0.0                                                        THE01125
      NZDIFF = NZDIFF- 1                                                  THE01130
 81   CONTINJE                                                            THE01135
C                                                                         THE01140
C     IF ALL TIME MEASUREMENTS EXCEED GATE BYPASS ADAPTIVE                THE01145
C     MANEUVER ROUTINE                                                    THE01150
C                                                                         THE01155
      IF (NZDIFF.EQ.0) GO TO 80                                           THE01160
      ZDIFAV=(ZDIFF(1)+ZDIFF(2)+ZDIFF(3)+ZDIFF(4))/NZDIFF                 THE01165
C                                                                         THE01170
C     DO NOT PERFORM ADAPTIVE MANEUVERING                                 THE01175
C     IF FILTER HAS NOT ACHIEVED STEADY STATE                             THE01180
C                                                                         THE01185
      IF(KK.LE.4)GO TO 80                                                 THE01190
C                                                                         THE01195
```

```
C      IF ZDIFAV MEETS CRITERIA TRANSFER OUT OF                          THE01200
C      ADAPTIVE MANEUVER ROUTINE                                         THE01205
C                                                                        THE01210
       IF(ZDIFAV.LE..10000D-04)GO TO 80                                  THE01215
C                                                                        THE01220
C      INCREASE THE GAIN                                                 THE01225
C                                                                        THE01230
       CALL QFIND(KK,SIGACC,SIGDIV,SIGCC,A14,Q)                          THE01235
       CALL ADD(PKK,Q,N,N,PKKM1)                                         THE01240
C                                                                        THE01245
C      PERFORM ADAPTIVE MANEUVERING BY REITERATING SAME TIME SLOT        THE01250
C                                                                        THE01255
       GO TO 711                                                         THE01260
C                                                                        THE01265
C                                                                        THE01270
  80   NZDIFF=4                                                          THE01275
       XDIFF1(KK)=XKK(1)-TRUX(KK)                                        THE01280
       XDIFF3(KK)=XKK(3)-TRUY(KK)                                        THE01285
       XDIFF5(KK)=XKK(5)-TRUZ(KK)                                        THE01290
C                                                                        THE01295
C        CALCULATE THE PREDICTIONS FOR PKKM1                             THE01300
C                                                                        THE01305
       CALL QFIND(KK,SIGACC,SIGDIV,SIGCC,A14,Q)                          THE01310
C      WRITE(6,100)                                                      THE01315
C      DO 765 I=1,N                                                      THE01320
C765    WRITE(6,766)(Q(I,J),J=1,N)                                       THE01325
C766    FORMAT(1X,5F10.4)                                                THE01330
C      WRITE(6,767)                                                      THE01335
C767    FORMAT(' ')                                                      THE01340
C                                                                        THE01345
         CALL PROD(PHI,PKK,N,N,N,PHIPKK)                                 THE01350
         CALL PROD(PHIPKK,PHIT,N,N,N,PKTEMP)                             THE01355
         CALL ADD(PKTEMP,Q,N,N,PKKM1)                                    THE01360
C                                                                        THE01365
C        CALCULATE NEW XKKM1                                             THE01370
C                                                                        THE01375
         CALL MMULT(PHI,XKK,N,N,XKKM1)                                   THE01380
          DO 41 IG=1,N                                                   THE01385
  41      XP(IG,KK)=XKK(IG)                                              THE01390
         DO 38 II=1,N                                                    THE01395
           DO 39 JJ=1,N                                                  THE01400
  39       P1(KK,II,JJ)=PKK(II,JJ)                                       THE01405
  38     CONTINUE                                                        THE01410
  99   CONTINUE                                                          THE01415
       WRITE(6,909)ZDM(1),ZDM(2),GXM(1),GXM(2),GYM(1),GYM(2)             THE01420
 909   FORMAT (6E14.5)                                                   THE01425
C                                                                        THE01430
C      SET UP ARRAYS AND SCALE FACTORS FOR PLOTP AND PLOTG               THE01435
```

```
C                                                                       THE01440
      C1(1)=285.                                                        THE01445
      C1(2)=330.                                                        THE01450
      C3(1)=0.                                                          THE01455
      C3(2)=500.                                                        THE01460
      C4(1)=0.                                                          THE01465
      C4(2)=40000.                                                      THE01470
      C5(1)=0.                                                          THE01475
      C5(2)=10000.                                                      THE01480
      DO 91 KK=1,ITIME                                                  THE01485
      X1(KK)=XP(1,KK)                                                   THE01490
      X3(KK)=XP(3,KK)                                                   THE01495
 91   X5(KK)=XP(5,KK)                                                   THE01500
           DO 92 IK=1,ITIME                                             THE01505
 92        KOUNT(IK)=IK                                                THE01510
              DO 113 IF=1,ITIME                                         THE01515
 113          AKOUNT(IF)=FLOAT(KOUNT(IF))                              THE01520
C                                                                       THE01525
C     GENERATE LINE PRINTER PLOTS                                       THE01530
C                                                                       THE01535
      DO 238 II=1,N                                                     THE01540
      JJ=II                                                             THE01545
        DO 239 KK=1,ITIME                                               THE01550
 239    PVEC(KK)=SQRT(P1(KK,II,JJ))                                     THE01555
        WRITE(6,900)                                                    THE01560
        CALL PLOTP(AKOUNT,PVEC,ITIME,0)                                 THE01565
 238   WRITE(6,800)II,JJ                                                THE01570
 800   FORMAT(23X,'PKK(',I1,',',I1,')','VS TIME')                       THE01575
 900   FORMAT('1')                                                      THE01580
      WRITE(6,900)                                                      THE01585
      CALL PLOTP(C3,C4,2,1)                                             THE01590
      CALL PLOTP(AKOUNT,X1,ITIME,2)                                     THE01595
      CALL PLOTP(AKOUNT,TRUX,ITIME,3)                                   THE01600
      WRITE(6,801)                                                      THE01605
 801  FORMAT(25X,'XPOSIT VS TIME')                                      THE01610
      WRITE(6,900)                                                      THE01615
      CALL PLOTP(AKOUNT,XDIFF1,ITIME,0)                                 THE01620
      WRITE(6,805)                                                      THE01625
 805  FORMAT(25X,'DEVIATION FROM TRUE X')                               THE01630
      WRITE(6,900)                                                      THE01635
      CALL PLOTP(C3,C5,2,1)                                             THE01640
      CALL PLOTP(AKOUNT,X3,ITIME,2)                                     THE01645
      CALL PLOTP(AKOUNT,TRUY,ITIME,3)                                   THE01650
      WRITE(6,802)                                                      THE01655
 802  FORMAT(25X,'YPOSIT VS TIME')                                      THE01660
      WRITE(6,900)                                                      THE01665
      CALL PLOTP(AKOUNT,XDIFF3,ITIME,0)                                 THE01670
      WRITE(6,806)                                                      THE01675
```

```
  806 FORMAT(25X,'DEVIATION FROM TRUE Y')                                THE01680
      WRITE(6,900)                                                       THE01685
      CALL PLOTP(C3,C1,2,1)                                              THE01690
      CALL PLOTP(AKOJNT,X5,ITIME,2)                                      THE01695
      CALL PLOTP(AKOJNT,TRUZ,ITIME,3)                                    THE01700
      WRITE(6,803)                                                       THE01705
  803 FORMAT(25X,'Z DEPTH VS TIME')                                      THE01710
      WRITE(6,900)                                                       THE01715
      CALL PLOTP(C4,C5,2,1)                                              THE01720
      CALL PLOTP(X1,X3,ITIME,2)                                          THE01725
      CALL PLOTP(TRUX,TRUY,ITIME,3)                                      THE01730
      WRITE(6,804)                                                       THE01735
  804 FORMAT(25X,'GEOGRAPHIC  Y VS X')                                   THE01740
      WRITE(6,900)                                                       THE01745
C     THIS IS THE BLOCK TO DO THE EXTRA PLOTS                            THE01750
      CALL PLOTP(C3,ZDM,2,1)                                             THE01755
      CALL PLOTP(AKOUNT,ZDIFF1(1,1),ITIME,2)                             THE01760
      CALL PLOTP(AKOJNT,ZDIFF1(1,2),ITIME,2)                             THE01765
      CALL PLOTP(AKOUNT,ZDIFF1(1,3),ITIME,2)                             THE01770
      CALL PLOTP(AKOUNT,ZDIFF1(1,4),ITIME,3)                             THE01775
      WRITE(6,70)                                                        THE01780
      WRITE(6,75)                                                        THE01785
      WRITE(6,900)                                                       THE01790
      CALL PLOTP(C3,GXM,2,1)                                             THE01795
      CALL PLOTP(AKOUNT,GX(1,1),ITIME,2)                                 THE01800
      CALL PLOTP(AKOJNT,GX(1,2),ITIME,2)                                 THE01805
      CALL PLOTP(AKOUNT,GX(1,3),ITIME,2)                                 THE01810
      CALL PLOTP(AKOUNT,GX(1,4),ITIME,3)                                 THE01815
      WRITE(6,76)                                                        THE01820
      WRITE(6,75)                                                        THE01825
      WRITE(6,900)                                                       THE01830
      CALL PLOTP(C3,GYM,2,1)                                             THE01835
      CALL PLOTP(AKOUNT,GY(1,1),ITIME,2)                                 THE01840
      CALL PLOTP(AKOUNT,GY(1,2),ITIME,2)                                 THE01845
      CALL PLOTP(AKOJNT,GY(1,3),ITIME,2)                                 THE01850
      CALL PLOTP(AKOUNT,GY(1,4),ITIME,3)                                 THE01855
      WRITE(6,78)                                                        THE01860
      WRITE(6,75)                                                        THE01865
      WRITE(6,900)                                                       THE01870
   70 FORMAT(20X,'ZDIFF VS TIME ')                                       THE01875
   75 FORMAT(2X,'TC=+, TX=*, TY=X, TZ=. ')                               THE01880
   76 FORMAT(20X,'GAIN IN X VS TIME ')                                   THE01885
   78 FORMAT(20X,'GAIN IN Y VS TIME ')                                   THE01890
C     END OF BLOCK FOR EXTRA PLOTS                                       THE01895
      STOP                                                               THE01900
      END                                                                THE01905
C                                                                        THE01910
C     GENERATE PLOTG CURVES                                              THE01915
```

```
C                                                                       THE01920
      II=1                                                              THE01925
      JJ=1                                                              THE01930
      DO 240 KK=1,ITIME                                                 THE01935
      PVEC(KK)=P1(KK,II,JJ)                                             THE01940
  240 PVEC(KK)=SQRT(PVEC(KK))                                           THE01945
      CALL PLOTG(AKOUNT,PVEC,ITIME,1,1,1,'TIME SLOT', 9,'RMS ESTIMATE OFTHE01950
     1 POSITION ERROR(FT)',34,0.,0.,0.,15.,12.,7.)                      THE01955
      II=3                                                              THE01960
      JJ=3                                                              THE01965
      DO 242 KK=1,ITIME                                                 THE01970
      PVEC(KK)=P1(KK,II,JJ)                                             THE01975
  242 PVEC(KK)=SQRT(PVEC(KK))                                           THE01980
      CALL PLOTG(AKOUNT,PVEC,ITIME,2,1,4,'TIME SLOT', 9,'RMS ESTIMATE OFTHE01985
     1 POSITION ERROR(FT)',34,0.,0.,0.,15.,12.,7.)                      THE01990
      II=2                                                              THE01995
      JJ=2                                                              THE02000
      DO 241 KK=1,ITIME                                                 THE02005
      PVEC(KK)=P1(KK,II,JJ)                                             THE02010
  241 PVEC(KK)=SQRT(PVEC(KK))                                           THE02015
      CALL PLOTG(AKOUNT,PVEC,ITIME,1,1,1,'TIME SLOT',9,'RMS ESTIMATE OF THE02020
     1VELOCITY ERROR(FT/SEC)',38,0.,0.,0.,0.,12.,7.)                    THE02025
      II=4                                                              THE02030
      JJ=4                                                              THE02035
      DO 243 KK=1,ITIME                                                 THE02040
      PVEC(KK)=P1(KK,II,JJ)                                             THE02045
  243 PVEC(KK)=SQRT(PVEC(KK))                                           THE02050
      CALL PLOTG(AKOUNT,PVEC,ITIME,2,1,4,'TIME SLOT',9,'RMS ESTIMATE OF THE02055
     1VELOCITY ERROR(FT/SEC)',38,0.,0.,0.,0.,12.,7.)                    THE02060
      II=5                                                              THE02065
      JJ=5                                                              THE02070
      DO 244 KK=1,ITIME                                                 THE02075
      PVEC(KK)=P1(KK,II,JJ)                                             THE02080
  244 PVEC(KK)=SQRT(PVEC(KK))                                           THE02085
      CALL PLOTG(AKOUNT,PVEC,ITIME,1,1,1,'TIME SLOT', 9,'RMS ESTIMATE OFTHE02090
     1 DEPTH ERROR(FT)',31,0.,0.,0.,15.,12.,7.)                         THE02095
      CALL PLOTG(AKOUNT,X1,ITIME,1,1,1,'TIME SLOT',9,'X-POSITION(FT)',  THE02100
     114,0.,0.,0.,0.,12.,7.)                                            THE02105
      CALL PLOTG(AKOUNT,TRUX,ITIME,2,1,4,'TIME SLOT',9,'X-POSITIOON(FT)'THE02110
     1,14,0.,0.,0.,0.,12.,7.)                                           THE02115
      CALL PLOTG(AKOUNT,XDIFF1,ITIME,1,1,1,'TIME SLOT',9,'ERROR IN POSITTHE02120
     1ION(FT)',21,0.,0.,-20.,20.,12.,7.)                                THE02125
      CALL PLOTG(AKOUNT,XDIFF3,ITIME,2,1,4,'TIME SLOT',9,'ERROR IN POSITTHE02130
     1ION(FT)',21,0.,0.,-20.,20.,12.,7.)                                THE02135
      CALL PLOTG(AKOUNT,X3,ITIME,1,1,1,'TIME SLOT',9,'Y-POSITION(FT)',14THE02140
     1,0.,0.,0.,0.,12.,7.)                                              THE02145
      CALL PLOTG(AKOUNT,TRUY,ITIME,2,1,4,'TIME SLOT',9,'Y-POSITION(FT)',THE02150
     114,0.,0.,0.,0.,12.,7.)                                            THE02155
```

```
 CALL PLOTG(AKOUNT,X5,ITIME,1,1,1,'TIME SLOT',9,'Z-DEPTH(FT)',11,  THE02160
10.,0.,0.,0.,12.,7.)                                               THE02165
 CALL PLOTG(AKOUNT,TRUZ,ITIME,2,1,4,'TIME SLOT',9,'Z-DEPTH(FT)',11,THE02170
10.,0.,0.,0.,12.,7.)                                               THE02175
 CALL PLOTG(AKOUNT,XDIFF5,ITIME,1,1,1,'TIME SLOT',9,'ERROR IN Z-DEPTHE02180
1TH(FT)',20,0.,0.,-20.,20.,12.,7.)                                 THE02185
 CALL PLOTG(TRUX,TRUY,ITIME,1,1,1,'X-POSITION(FT)',14,'Y-POSITION(FTHE02190
1T)',14,0.,0.,0.,0.,12.,7.)                                        THE02195
 CALL PLOTG(C3,ZDM,2,1,0,0,'TIME SLOT',9,'TIME RESIDUALS(SEC)',19, THE02200
10.,0.,0.,0.,12.,7.)                                               THE02205
 CALL PLOTG(AKOUNT,ZDIFFI(1,1),ITIME,2,1,1,'TIME SLOT',9,'TIME RESITHE02210
1DUALS(SEC)',19,0.,0.,0.,0.,12.,7.)                                THE02215
 CALL PLOTG(AKOUNT,ZDIFFI(1,2),ITIME,3,1,2,'TIME SLOT',9,'TIME RESITHE02220
1DUALS(SEC)',19,0.,0.,0.,0.,12.,7.)                                THE02225
 CALL PLOTG(AKOUNT,ZDIFFI(1,3),ITIME,4,1,3,'TIME SLOT',9,'TIME RESITHE02230
1DUALS(SEC)',19,0.,0.,0.,0.,12.,7.)                                THE02235
 CALL PLOTG(AKOUNT,ZDIFFI(1,4),ITIME,5,1,4,'TIME SLOT',9,'TIME RESITHE02240
1DUALS(SEC)',19,0.,0.,0.,0.,12.,7.)                                THE02245
 CALL PLOTG(C3,GXM,2,1,0,0,'TIME SLOT',9,'X-GAIN',6,0.,0.,0.,0.,   THE02250
112.,7.)                                                           THE02255
 CALL PLOTG(AKOUNT,GX(1,1),ITIME,2,1,1,'TIME SLOT',9,'X-GAIN',6,0.,THE02260
10.,0.,0.,12.,7.)                                                  THE02265
 CALL PLOTG(AKOUNT,GX(1,2),ITIME,3,1,2,'TIME SLOT',9,'X-GAIN',6,0.,THE02270
10.,0.,0.,12.,7.)                                                  THE02275
 CALL PLOTG(AKOUNT,GX(1,3),ITIME,4,1,3,'TIME SLOT',9,'X-GAIN',6,0.,THE02280
10.,0.,0.,12.,7.)                                                  THE02285
 CALL PLOTG(AKOUNT,GX(1,4),ITIME,5,1,4,'TIME SLOT',9,'X-GAIN',6,0.,THE02290
10.,0.,0.,12.,7.)                                                  THE02295
 CALL PLOTG(C3,GYM,2,1,0,0,'TIME SLOT',9,'Y-GAIN',6,0.,0.,0.,0.,   THE02300
112.,7.)                                                           THE02305
 CALL PLOTG(AKOUNT,GY(1,1),ITIME,2,1,1,'TIME SLOT',9,'Y-GAIN',6,0.,THE02310
10.,0.,0.,12.,7.)                                                  THE02315
 CALL PLOTG(AKOUNT,GY(1,2),ITIME,3,1,2,'TIME SLOT',9,'Y-GAIN',6,0.,THE02320
10.,0.,0.,12.,7.)                                                  THE02325
 CALL PLOTG(AKOUNT,GY(1,3),ITIME,4,1,3,'TIME SLOT',9,'Y-GAIN',6,0.,THE02330
10.,0.,0.,12.,7.)                                                  THE02335
 CALL PLOTG(AKOUNT,GY(1,4),ITIME,5,1,4,'TIME SLOT',9,'Y-GAIN',6,0.,THE02340
10.,0.,0.,12.,7.)                                                  THE02345
 CALL PLOT(0.0,0.0,999)                                            THE02350
 STOP                                                              THE02355
 END                                                               THE02360
```

```
      SUBROUTINE TRAJEC(K,DATA,ZI,TD)                                    THE02365
C                                                                        THE02370
C     SUBROUTINE COMPUTES TRUE TRAJECTORY OF TORPEDO                     THE02375
C  DATA(1)=TRUE X POSITION,DATA(2)=TRUE Y POSITION                       THE02380
C  DATA(3)=TRUE Z POSITION,DATA(4)=TRUE X VELOCITY                       THE02385
C  DATA(5)=TRUE Y VELOCITY,DATA(6)=0.,DATA(7)=0.,DATA(8)=0.              THE02390
C  DATA(9)=0.,DATA(10)=0.,DATA(11)=0.                                    THE02395
C  DATA(12)=ACCELLERATION ANGLE(RADIANS),DATA(13)=TURN RATE(RAD./SEC)    THE02400
C  DATA(14)=SAMPLE TIME,DATA(15)=HORIZONTAL ACCELLERATION(FT/SEC)        THE02405
C  DATA(16), DATA(17)=TIME SLOT INTERVAL DURING WHICH MANEUVER           THE02410
C     TAKES PLACE,ZI=TRUE TIMES,TD=TRUE POSITION                         THE02415
C                                                                        THE02420
      DOUBLE PRECISION DATA,ZI                                           THE02425
      DIMENSION DATA(23),ZI(4),TD(3)                                     THE02430
      T=0.0                                                              THE02435
      VEL=4860.                                                          THE02440
      RANGE=DSQRT(DATA(1)*DATA(1)+DATA(2)*DATA(2)+DATA(3)*DATA(3))       THE02445
      I = 1                                                              THE02450
      ZI(I)=1./VEL*(((DATA(1)+15.)**2)+((DATA(2)+15.)**2)+((DATA(3)      THE02455
     1     +15.)**2))**.5                                                THE02460
      I = 2                                                              THE02465
      ZI(I)=1./VEL*(((DATA(1)-15.)**2)+((DATA(2)+15.)**2)+((DATA(3)      THE02470
     1     +15.)**2))**.5                                                THE02475
      I = 3                                                              THE02480
      ZI(I)=1./VEL*(((DATA(1)+15.)**2)+((DATA(2)-15.)**2)+((DATA(3)      THE02485
     1     +15.)**2))**.5                                                THE02490
      I = 4                                                              THE02495
      ZI(I)=1./VEL*(((DATA(1)+15.)**2)+((DATA(2)+15.)**2)+((DATA(3)      THE02500
     1     -15.)**2))**.5                                                THE02505
      DO 9 I=1,3                                                         THE02510
      TD(I)=DATA(I)                                                      THE02515
 9    CONTINUE                                                           THE02520
      IF((K.LE.DATA(17)).AND.(K.GT. DATA(16))) GO TO 11                  THE02525
      DATA(7)=0.0                                                        THE02530
      DATA(8)=0.0                                                        THE02535
      DATA(14)=1.31                                                      THE02540
      GO TO 12                                                           THE02545
 11   DATA(14)=.005                                                      THE02550
 13   DATA(12)=DATA(12)+DATA(13)*DATA(14)                                THE02555
      DATA(7)=DATA(15)*DCOS(DATA(12))                                    THE02560
      DATA(8)=DATA(15)*DSIN(DATA(12))                                    THE02565
 12   DO 10 I=1,5                                                        THE02570
      DATA(I)=DATA(I)+DATA(I+3)*DATA(14)+(((DATA(14))**2)/2)*DATA(I+6)   THE02575
 10   CONTINUE                                                           THE02580
      T=T+DATA(14)                                                       THE02585
      IF(ABS(T-1.31).LE.0.0001) RETURN                                   THE02590
      GO TO 13                                                           THE02595
      END                                                                THE02600
```

```
      SUBROUTINE TRAJC3(K,DATA,ZI,TD,XB,YB,ZB)                          THE02605
C                                                                       THE02610
C     SUBROUTINE COMPUTES TRUE TRAJECTORY OF TORPEDO                    THE02615
C                                                                       THE02620
C  DATA(1)=TRUE X POSITION,DATA(2)=TRUE Y POSITION                      THE02625
C  DATA(3)=TRUE Z POSITION,DATA(4)=TRUE X VELOCITY                      THE02630
C  DATA(5)=TRUE Y VELOCITY,DATA(6)=0.,DATA(7)=0.,DATA(8)=0.             THE02635
C  DATA(9)=0.,DATA(10)=0.,DATA(11)=0.                                   THE02640
C  DATA(12)=ACCELLERATION ANGLE(RADIANS),DATA(13)=TURN RATE(RAD./SEC)   THE02645
C  DATA(14)=SAMPLE TIME,DATA(15)=HORIZONTAL ACCELLERATION(FT/SEC)       THE02650
C  DATA(16), DATA(17)=TIME SLOT INTERVAL DURING WHICH MANEUVER          THE02655
C     TAKES PLACE,ZI=TRUE TIMES,TD=TRUE POSITION                        THE02660
C                                                                       THE02665
      DOUBLE PRECISION DATA,ZI                                          THE02670
      DIMENSION DATA(23),ZI(4),TD(3),XB(4),YB(4),ZB(4)                  THE02675
      T=0.0                                                             THE02680
      VEL=4860.                                                         THE02685
      RANGE=DSQRT(DATA(1)*DATA(1)+DATA(2)*DATA(2)+DATA(3)*DATA(3))      THE02690
      DO 5 I=1,4                                                        THE02695
         ZI(I)=1./VEL*(((DATA(1)-XB(I))**2)+((DATA(2)-YB(I))**2)        THE02700
     1       +((DATA(3)-ZB(I))**2))**0.5                                THE02705
    5 CONTINUE                                                          THE02710
      DO 9 I=1,3                                                        THE02715
      TD(I)=DATA(I)                                                     THE02720
    9 CONTINUE                                                          THE02725
      IF((K.LE.DATA(17)).AND.(K.GT. DATA(16))) GO TO 11                 THE02730
      DATA(7)=0.0                                                       THE02735
      DATA(8)=0.0                                                       THE02740
      DATA(14)=1.31                                                     THE02745
      GO TO 12                                                          THE02750
   11 DATA(14)=.005                                                     THE02755
   13 DATA(12)=DATA(12)+DATA(13)*DATA(14)                               THE02760
      DATA(7)=DATA(15)*DCOS(DATA(12))                                   THE02765
      DATA(8)=DATA(15)*DSIN(DATA(12))                                   THE02770
   12 DO 10 I=1,5                                                       THE02775
      DATA(I)=DATA(I)+DATA(I+3)*DATA(14)+(((DATA(14))**2)/2)*DATA(I+6)  THE02780
   10 CONTINUE                                                          THE02785
      T=T+DATA(14)                                                      THE02790
      IF(ABS(T-1.31).LE.0.0001) RETURN                                  THE02795
      GO TO 13                                                          THE02800
      END                                                               THE02805
```

```
      SUBROUTINE INIT (XKKM1,PKKM1)                                     THE02810
C                                                                       THE02815
C THIS ROUTINE IS TO INITIALIZE THE ARRAYS XKKM1 AND PKKM1.             THE02820
C THESE VARIABLES ARE PART OF A COMMON BLOCK.                           THE02825
C                                                                       THE02830
      DOUBLE PRECISION XKKM1,PKKM1                                      THE02835
      DIMENSION XKKM1(5),PKKM1(5,5)                                     THE02840
      DO 20 J=1,5                                                       THE02845
         DO 10 I=1,5                                                    THE02850
            PKKM1(I,J)=0.0                                              THE02855
   10 CONTINUE                                                          THE02860
      XKKM1(1)=38975.                                                   THE02865
      XKKM1(2)=-50.                                                     THE02870
      XKKM1(3)=6025.                                                    THE02875
      XKKM1(4)=-10.                                                     THE02880
      XKKM1(5)=300.                                                     THE02885
   20 CONTINUE                                                          THE02890
      DO 30 I=1,5                                                       THE02895
         PKKM1(I,I)=1000.                                               THE02900
   30 CONTINUE                                                          THE02905
      RETURN                                                            THE02910
      END                                                               THE02915
```

```
      SUBROUTINE CZHAT(I,ZHAT)                                          THE02920
C                                                                       THE02925
CC    THIS SUBROUTINE COMPUTES ESTIMATES OF TRANSIT TIME MEASUREMENTS   THE02930
CC    FOR SINGLE ARRAY TRACKING                                         THE02935
C                                                                       THE02940
      DOUBLE PRECISION XKKM1,ZHAT                                       THE02945
      COMMON XKKM1(5)                                                   THE02950
      VEL=4860.                                                         THE02955
      IF(I.EQ.1)ZHAT=1./VEL*(((XKKM1(1)+15.)**2)+((XKKM1(3)+15.)**2)    THE02960
     1+((XKKM1(5)+15.)**2))**0.5                                        THE02965
      IF(I.EQ.2)ZHAT=1./VEL*(((XKKM1(1)-15.)**2)+((XKKM1(3)+15.)**2)    THE02970
     1+((XKKM1(5)+15.)**2))**0.5                                        THE02975
      IF(I.EQ.3)ZHAT=1./VEL*(((XKKM1(1)+15.)**2)+((XKKM1(3)-15.)**2)    THE02980
     1+((XKKM1(5)+15.)**2))**0.5                                        THE02985
      IF(I.EQ.4)ZHAT=1./VEL*(((XKKM1(1)+15.)**2)+((XKKM1(3)+15.)**2)    THE02990
     1+((XKKM1(5)-15.)**2))**0.5                                        THE02995
      RETURN                                                            THE03000
      END                                                               THE03005
```

```
      SUBROUTINE CZHAT3 (I,ZHAT,XB,YB,ZB)                                   THE03010
C                                                                           THE03015
C     THIS SUBROUTINE COMPUTES ESTIMATES OF TRANSIT TIME MEASUREMENTS       THE03020
C     FOR MULTIPLE ARRAY TRACKING                                           THE03025
C                                                                           THE03030
      DOUBLE PRECISION XKKM1,ZHAT                                           THE03035
      COMMON XKKM1(5)                                                       THE03040
      DIMENSION XB(4),YB(4),ZB(4)                                           THE03045
      VEL=4860.                                                             THE03050
      XO=XB(I)                                                              THE03055
      YO=YB(I)                                                              THE03060
      ZO=ZB(I)                                                              THE03065
      ZHAT=(1./VEL)*(((XKKM1(1)-XO)**2)+((XKKM1(3)-YO)**2)+((XKKM1(5)       THE03070
     1-ZO)**2))**0.5                                                        THE03075
      RETURN                                                                THE03080
      END                                                                   THE03085
```

```
      SUBROUTINE CHROW(I,HROW)                                          THE03090
C                                                                       THE03095
C     THIS SUBROUTINE USED FOR SINGLE ARRAY TRACKING                    THE03100
C                                                                       THE03105
      DOUBLE PRECISION XKKM1,DENOM1,DENOM2,DENOM3,DENOM4,DENOM,HROW,    THE03110
     1H1,H3,H5                                                          THE03115
      COMMON XKKM1(5)                                                   THE03120
      DIMENSION HROW(5)                                                 THE03125
      VEL=4860.                                                         THE03130
      DENOM1=(((XKKM1(1)+15.)**2)+((XKKM1(3)+15.)**2)+((XKKM1           THE03135
     1(5)+15.)**2))**0.5                                                THE03140
      DENOM2=(((XKKM1(1)-15.)**2)+((XKKM1(3)+15.)**2)+((XKKM1           THE03145
     1(5)+15.)**2))**0.5                                                THE03150
      DENOM3=(((XKKM1(1)+15.)**2)+((XKKM1(3)-15.)**2)+((XKKM1           THE03155
     1(5)+15.)**2))**0.5                                                THE03160
      DENOM4=(((XKKM1(1)+15.)**2)+((XKKM1(3)+15.)**2)+((XKKM1           THE03165
     1(5)-15.)**2))**0.5                                                THE03170
      A1=1.                                                             THE03175
      A2=1.                                                             THE03180
      A3=1.                                                             THE03185
      DENOM=DENOM1                                                      THE03190
      IF(I.EQ.2)DENOM=DENOM2                                            THE03195
      IF(I.EQ.3)DENOM=DENOM3                                            THE03200
      IF(I.EQ.4)DENOM=DENOM4                                            THE03205
      IF(I.EQ.2)A1=-1.                                                  THE03210
      H1=(1./VEL)*((XKKM1(1)+A1*15.)/DENOM)                             THE03215
      IF(I.EQ.3)A2=-1.                                                  THE03220
      H3=(1./VEL)*((XKKM1(3)+A2*15.)/DENOM)                             THE03225
      IF(I.EQ.4)A3=-1.                                                  THE03230
      H5=(1./VEL)*((XKKM1(5)+A3*15.)/DENOM)                             THE03235
      HROW(1)=H1                                                        THE03240
      HROW(3)=H3                                                        THE03245
      HROW(5)=H5                                                        THE03250
      DO 27 J=2,4,2                                                     THE03255
27    HROW(J)=0.                                                        THE03260
      RETURN                                                            THE03265
      END                                                               THE03270
```

```
      SUBROUTINE CHROW3 (I,HROW,XB,YB,ZB)                                THE03275
C                                                                        THE03280
CC    THIS SUBROUTINE USED FOR MULTIPLE ARRAY TRACKING                   THE03285
C                                                                        THE03290
      DOUBLE PRECISION XKKM1,HROW,DENOM                                  THE03295
      COMMON XKKM1(5)                                                    THE03300
      DIMENSION HROW(5),XB(4),YB(4),ZB(4)                                THE03305
      VEL=4860.                                                          THE03310
      XO=XB(I)                                                           THE03315
      YO=YB(I)                                                           THE03320
      ZO=ZB(I)                                                           THE03325
      DENOM=(((XKKM1(1)-XO)**2)+((XKKM1(3)-YO)**2)+((XKKM1(5)-ZO)        THE03330
     1**2))**0.5                                                         THE03335
      HROW(1)=(1./VEL)*((XKKM1(1)-XO)/DENOM)                             THE03340
      HROW(3)=(1./VEL)*((XKKM1(3)-YO)/DENOM)                             THE03345
      HROW(5)=(1./VEL)*((XKKM1(5)-ZO)/DENOM)                             THE03350
      DO 27 J=2,4,2                                                      THE03355
   27   HROW(J)=0.                                                       THE03360
      RETURN                                                             THE03365
      END                                                                THE03370
```

```
      SUBROUTINE QFIND(K,SIGACC,SIGDIV,SIGCC,A,Q)                       THE03375
C                                                                       THE03380
CC    THIS SUBROUTINE COMPUTES THE ADAPTIVE Q MATRIX                    THE03385
C                                                                       THE03390
      DOUBLE PRECISION XHKKM1,PKKM1,PKK,XHKK,Q                          THE03395
      COMMON XHKKM1(5),PKKM1(5,5),PKK(5,5),XHKK(5)                      THE03400
      DIMENSION Q(5,5)                                                  THE03405
      IF(K.NE.1) GO TO 15                                               THE03410
      DO 10 I=1,5                                                       THE03415
           DO 10 J=1,5                                                  THE03420
   10  Q(I,J)=0.0                                                       THE03425
      SIGACC=SIGACC**2                                                  THE03430
      Q(5,5)=(SIGDIV*A)**2                                              THE03435
      SIGCC=SIGCC**2                                                    THE03440
      G1=(A**2)/2.0                                                     THE03445
      G2=G1**2                                                          THE03450
      G3=A*G1                                                           THE03455
      A2=A**2                                                           THE03460
   15  A1=XHKK(2)**2+XHKK(4)**2                                         THE03465
      A3=XHKK(2)/SQRT(A1)                                               THE03470
      B=XHKK(4)                                                         THE03475
      C=XHKK(4)/SQRT(A1)                                                THE03480
      D=XHKK(2)                                                         THE03485
      E1=(A3**2)*SIGACC+(B**2)*SIGCC                                    THE03490
      E12=A3*C*SIGACC-B*D*SIGCC                                         THE03495
      E2=(C**2)*SIGACC+(D**2)*SIGCC                                     THE03500
      Q(1,1)=E1*G2                                                      THE03505
      Q(1,2)=G3*E1                                                      THE03510
      Q(1,3)=E12*G2                                                     THE03515
      Q(1,4)=G3*E12                                                     THE03520
      Q(2,2)=A2*E1                                                      THE03525
      Q(2,3)=G3*E12                                                     THE03530
      Q(2,4)=A2*E12                                                     THE03535
      Q(3,3)=G2*E2                                                      THE03540
      Q(3,4)=G3*E2                                                      THE03545
      Q(4,4)=A2*E2                                                      THE03550
      DO 27 I=1,4                                                       THE03555
           DO 27 J=1,I                                                  THE03560
   27  Q(I,J)=Q(J,I)                                                    THE03565
      RETURN                                                            THE03570
      END                                                               THE03575
```

```
      SUBROUTINE PROD(A,B,N,M,L,C)                                      THE03580
      DOUBLE PRECISION A,B,C                                            THE03585
      DIMENSION A(N,M),B(M,L),C(N,L)                                    THE03590
      DO 1 I=1,N                                                        THE03595
      DO 1 J=1,L                                                        THE03600
    1 C(I,J)=0.                                                         THE03605
      DO 2 I=1,N                                                        THE03610
      DO 2 J=1,L                                                        THE03615
      DO 2 K=1,M                                                        THE03620
    2 C(I,J)=C(I,J)+A(I,K)*B(K,J)                                       THE03625
      RETURN                                                            THE03630
      END                                                               THE03635
```

```
      SUBROUTINE MMJLT(A,B,N,M,C)                                       THE03640
      DOUBLE PRECISION A,B,C                                            THE03645
      DIMENSION A(N,M),B(M),C(N)                                        THE03650
      DO 3 I=1,N                                                        THE03655
      C(I)=0.                                                           THE03660
      DO 4 J=1,M                                                        THE03665
    4 C(I)=C(I)+A(I,J)*B(J)                                             THE03670
    3 CONTINUE                                                          THE03675
      RETURN                                                            THE03680
      END                                                               THE03685
```

```
      SUBROUTINE VMJLT(A,B,N,C)                                         THE03690
      DOUBLE PRECISION A,B                                              THE03695
      DIMENSION A(N),B(N)                                               THE03700
      C=0.                                                              THE03705
      DO 6 I=1,N                                                        THE03710
    6 C=C+A(I)*B(I)                                                     THE03715
      RETURN                                                            THE03720
      END                                                               THE03725
```

```
      SUBROUTINE TRANS(A,N,M,B)                                        THE03730
      DOUBLE PRECISION A,B                                             THE03735
      DIMENSION A(N,M),B(M,N)                                          THE03740
      DO 13 I=1,N                                                      THE03745
      DO 13J=1,M                                                       THE03750
   13 B(J,I)=A(I,J)                                                    THE03755
      RETURN                                                           THE03760
      END                                                              THE03765
```

```
      SUBROUTINE ADD (A,B,N,M,C)                                        THE03770
      DOUBLE PRECISION A,B,C                                            THE03775
      DIMENSION A(N,M),B(N,M),C(N,M)                                    THE03780
      DO 15 I=1,N                                                       THE03785
      DO 15 J=1,M                                                       THE03790
   15 C(I,J)=A(I,J)+B(I,J)                                              THE03795
      RETURN                                                            THE03800
      END                                                               THE03805
```

LIST OF REFERENCES

1. Benson, Eric J. An Application of Kalman Filtering to Underwater Tracking, Master's Thesis, Naval Postgraduate School, December 1976.

2. Dwyer, Dennis M. Real Time Kalman Filtering For Torpedo Range Tracking, Master's Thesis, Naval Postgraduate School, December 1978.

3. Technical Manual, NAVORD OD 41964, NAVTOPRSTA Keyport Range Complex and Associated Data, May 1970.

4. Kirk, Donald E., EE4413 Optimal Control Systems Class Notes, Unpublished.

5. Mitschang, George W. An Application of Nonlinear Filtering Theory to Passive Target Location and Tracking. Ph.D. Thesis, Naval Postgraduate School, 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor H. A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
5.	Associate Professor Alex Gerba, Code 62Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Commanding Officer Naval Torpedo Station Keyport, Washington 98345	5
7.	Paul A. O'Brien ASATD-AT-51 Naval Air Test Center Patuxent River, Maryland 20670	2


DATE
FILMED
8